



UNIVERSIDAD ESAN

FACULTAD DE INGENIERÍA

INGENIERÍA DE TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

Diseño de un sistema de visión computacional para el pre-diagnóstico de la Enfermedad de Parkinson a partir de la escritura de una persona

Tesis para optar el Título de Ingeniera de Tecnologías de Información y Sistemas que presenta:

Virginia del Pilar Monroy Malca – DNI 74043491

Asesor: Joseph Ballón Alvarez – DNI 02441461

Lima, marzo de 2021

Esta tesis denominada:

DISEÑO DE UN SISTEMA DE VISIÓN COMPUTACIONAL PARA EL PRE-
DIAGNÓSTICO DE LA ENFERMEDAD DE PARKINSON A PARTIR DE LA
ESCRITURA DE UNA PERSONA

ha sido aprobada.



.....
Marks Arturo Calderón Niquin (Jurado Presidente)



.....
Wilfredo Mamani Ticona (Jurado)



.....
Pedro Nelson Shiguihara Juarez (Jurado)

Universidad ESAN

2021

Diseño de un sistema de visión computacional para el pre-diagnóstico de la Enfermedad de Parkinson a partir de la escritura de una persona

DEDICATORIA

Dedico esta tesis a DIOS, quien inspiró mi espíritu para la conclusión de esta tesis de grado. A mis padres, hermana, abuelos, tía, primos y enamorado que siempre me dan su apoyo y consejos. A mi asesor de tesis, a mis compañeros de universidad y trabajo, a mis maestros y amigos, quienes siempre confiaron en mí. A todos ellos les agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

Índice

Introducción	3
Capítulo I: Planteamiento del Problema	5
1.1 Descripción de la Realidad Problemática	5
1.2 Formulación del Problema	10
1.2.1 problema general	10
1.2.2 problemas específicos	10
1.3 Objetivos de la Investigación.....	10
1.3.1 objetivo general.....	10
1.3.2 objetivos específicos.....	11
1.4 Justificación de la Investigación.....	11
1.4.1 teórica	11
1.4.2 práctica	12
1.4.3 metodológica.....	12
1.5 Delimitación del Estudio	13
1.5.1 espacial.....	13
1.5.2 temporal.....	13
1.5.3 conceptual.....	13
Capítulo II: Marco Teórico	14
2.1 Antecedentes de la Investigación	14
2.2 Bases Teóricas.....	25
2.2.1 enfermedad de parkinson	25
2.2.2 visión computacional	28
2.2.3 inteligencia artificial	47
2.3 Marco Conceptual o Contexto de la Investigación.....	76
2.3.1 inteligencia artificial	76
2.3.2 machine learning.....	76
2.3.3 aprendizaje supervisado.....	76
2.3.4 support vector machine.....	76
2.3.5 redes neuronales	76
2.3.6 regresión logística	76
2.3.7 árbol de decisiones	77
2.3.8 random forest.....	77

2.3.9 trastorno	77
2.3.10 degeneración.....	77
2.3.11 síntoma.....	77
2.3.12 neurotransmisor.....	77
2.3.13 dopamina	78
2.4 Hipótesis	78
2.4.1 hipótesis general.....	78
2.4.2 hipótesis específicas	78
Capítulo 3: Metodología	79
3.1 Diseño, tipo y enfoque de Investigación	79
3.1.1 diseño	79
3.1.1 tipo – nivel	79
3.1.2 enfoque.....	79
3.2 Población y Muestra.....	80
3.3 Operacionalización de Variables	82
3.4 Técnicas de Recolección de Datos	84
3.5 Técnicas para el Procesamiento y Análisis de la Información	86
3.5.1 metodología de la implementación de la solución	86
3.5.2 metodología para la medición de resultados	127
3.6 Cronograma de Actividades.....	130
3.7 Presupuesto	132
Capítulo 4: Desarrollo del Experimento	133
4.1 Adquisición del Dataset	133
4.2 Fase de Preprocesamiento	145
4.3 Extracción de características.....	149
4.4 Modelo y Clasificación	152
4.4.1 metodología 1	152
4.4.2 metodología 2	153
4.4.3 metodología 3	157
Capítulo 5: Análisis y Discusión de Resultados.....	158
5.1 Resultados Base de Datos de Brasil	159
5.2 Resultados Base de Datos de Perú	174
5.3 Despliegue	189
5.3.1 arquitectura para el usuario	190
5.3.2 arquitectura para el miembro de la asociación.....	207

Capítulo 6: Conclusiones y Recomendaciones	211
6.1 Conclusiones	211
6.2 Recomendaciones	213
Referencias.....	215
Anexos	221

Índice de Figuras

Figura 1. Cantidad de personas diagnosticadas con la Enfermedad de Parkinson en el mundo según la OMS y PARKINSON'S NEWS TODAY.	6
Figura 2. Cantidad de personas diagnosticadas con la Enfermedad de Parkinson en el Perú según Instituto Nacional de Ciencias Neurológicas (INCN).	7
Figura 3. Reconocimiento óptico de caracteres	9
Figura 4. Modelo de prueba escrita la cual fue entregada a los pacientes	15
Figura 5. Pluma especial que capta a través de señales los movimientos de la mano y señales captadas por la pluma.....	16
Figura 6. Resumen de la metodología usada en el presente artículo.	19
Figura 7. Resumen de la extracción de características.	20
Figura 8. Imagen dividida entre el trazo real y el del paciente	22
Figura 9. Procedimiento del Preprocesamiento	23
Figura 10. Estructura de la Red Convolutiva	24
Figura 11. Sustancia negra en la EP.....	27
Figura 12. Agente transmisor en la EP	27
Figura 13. Ejemplo de escritura de paciente que presenta la EP	28
Figura 14. Ejemplos de Aplicaciones de Visión Computacional	29
Figura 15. Características de una imagen	31
Figura 16. Técnicas de Segmentación de Imágenes	32
Figura 17. Extracción de características	33
Figura 18. Clasificación de objetos.....	34
Figura 19. Detección de Rostros.....	35
Figura 20. Ejemplo de aumento lineal de contraste	36
Figura 21. Histograma de dos imágenes	37
Figura 22. Imagen real vs Imagen con Filtro de la media aritmética	37
Figura 23. Aplicación del filtro de la mediana de una imagen	38
Figura 24. Máscara para filtro Gaussiano	38
Figura 25. Imagen real vs Imagen con Filtro Gaussiano	39
Figura 26. Gaussian vs Diferencia de Gaussian.....	40
Figura 27. Selección de puntos característicos	41
Figura 28. Descriptor de un punto clave.....	42
Figura 29. Respuestas de la Haar-wavelet	43

Figura 30. Propiedades del descriptor para tres patrones de intensidad de imagen.....	44
Figura 31. Resultado coincidente utilizando ORB en imágenes del mundo real con cambio de punto de vista.	45
Figura 32. Metodología del HOG	46
Figura 33. Ejemplo de HOG	47
Figura 34. Interacción entre los agentes y entornos.....	49
Figura 35. Acciones de una aspiradora con solo dos entornos	49
Figura 36. Números del 0 a 9 escritos a mano.....	50
Figura 37. Clusters resultantes al aplicar K-Means	51
Figura 38. Datasets linealmente no separables	54
Figura 39. Datasets linealmente separables	54
Figura 40. Modelo no lineal de una neurona	55
Figura 41. Gráfica del Perceptron.....	55
Figura 42. Arquitectura de un Perceptron Multilayer.....	56
Figura 43. Actualización de pesos	58
Figura 44. Función Sigmoidal.....	58
Figura 45. Función de Gauss	59
Figura 46. Función Escalar	60
Figura 47. Función Tangente Hiperbólica	60
Figura 48. Representaciones de redes profundas entrenadas.....	61
Figura 49. Características de una Red Neuronal Convolucional	62
Figura 50. ¿Cómo funciona una convolución?	63
Figura 51. Imagen real e imagen con filtro.....	64
Figura 52. Ejemplo de Max Pooling.....	64
Figura 53. Ejemplo de Dropout	65
Figura 54. Ejemplo de Fully Connected	66
Figura 55. Ejemplo de Fully Connected	66
Figura 56. Arquitectura de VGG16	72
Figura 57. Arquitectura de AlexNet.....	73
Figura 58. Arquitectura de GoogLeNet	73
Figura 59. Error de entrenamiento (izquierda) y error de prueba (derecha) en CIFAR-10 con redes “planas” de 20 y 56 capas.....	74
Figura 60. Conexiones saltadas.....	75
Figura 61. Arquitectura de ResNet	76

Figura 62. Primera hoja del examen de escritura.....	84
Figura 63. Primera hoja del examen de escritura.....	85
Figura 64. Metodología de la investigación.....	86
Figura 65. Ejemplos de espirales y meandros de la prueba de Brasil.....	88
Figura 66. Ejemplos de espirales y meandros de la prueba de Brasil.....	89
Figura 67. Redimensión de imágenes	93
Figura 68. Función del algoritmo que realiza los cortes.....	95
Figura 69. Pseudocódigo del algoritmo que realiza los cortes.....	96
Figura 70. Arquitectura de BOVW	99
Figura 71. Agrupación de los descriptores con K-Means.....	99
Figura 72. Pseudocódigo del algoritmo BOVW	100
Figura 73. Keypoints en la imagen con SIFT	101
Figura 74. Keypoints en la imagen con SURF	102
Figura 75. Keypoints en la imagen con ORB	103
Figura 76. Diagrama de proceso de HOG.....	103
Figura 77. Pseudocódigo de la asignación de target.....	105
Figura 78. Pseudocódigo de la función de desordenamiento.....	106
Figura 79. Pseudocódigo de la función train_test_val	107
Figura 80. Proceso de la división del dataset en train, test y val	107
Figura 81. Funcionamiento de SVM.....	108
Figura 82. Pseudocódigo de la función GridSearchCV	109
Figura 83. Iteraciones con GridSearch.....	109
Figura 84. k-Fold Cross Validation	110
Figura 85. Pseudocódigo de la función train_test_val_imagenes.....	112
Figura 86. Transfer Learning VGG16	114
Figura 87. Arquitectura final del modelo de VGG16	115
Figura 88. Transfer Learning VGG19	116
Figura 89. Arquitectura final del modelo de VGG19	117
Figura 90. Transfer Learning Inception	118
Figura 91. Arquitectura final del modelo de Inception.....	119
Figura 92. Transfer Learning Resnet	120
Figura 93. Arquitectura final del modelo de LeNet.....	121
Figura 94. Proceso de entrenamiento de una CNN.....	121
Figura 95. Proceso de entrenamiento de una CNN.....	123

Figura 96. Ejemplo de aplicación de KNN	124
Figura 97. Creación de conjuntos de datos individuales.....	125
Figura 98. Selección aleatoria de características para el árbol de construcción	126
Figura 99. Predicciones usando un bosque aleatorio	126
Figura 100. Medición de resultados de implementación	127
Figura 101. Curva ROC	129
Figura 102. Ejemplos de espiral y meandro.....	134
Figura 103. Noticia diario Correo sobre EP.....	135
Figura 104. Conversación con Christine por Whatsapp	136
Figura 105. Correo sobre los nombres y apellidos de miembros de la Asociación.....	137
Figura 106. Proceso de armado de los sobres	138
Figura 107. Fila de sobres finales	139
Figura 108. Ejemplo de entrega de sobres	140
Figura 109. Ejemplo de recojo de sobres.....	141
Figura 110. Recojo de sobre con Christine Jeyachandran	141
Figura 111. Desinfección de los sobres	142
Figura 112. Modelo de celular Moto G9 Plus que se utilizó para tomar foto de los exámenes de escritura	143
Figura 113. Trípode con el celular para tomar las fotos	144
Figura 114. Ejemplo del examen de escritura.....	145
Figura 115. Ejemplos de imágenes generadas con Data Augmentation – Dataset Brasil	145
Figura 116. Imagen de un meandro en escala de grises.....	146
Figura 117. Ejemplos de imágenes que fueron cortadas del examen de escritura – Dataset Perú	146
Figura 118. Primer preprocesamiento aplicado a las imágenes – Dataset Perú.....	147
Figura 119. Métricas obtenidas del primer preprocesamiento aplicado a las imágenes del segundo dataset	147
Figura 120. Segundo preprocesamiento aplicado a las imágenes – Dataset Perú	148
Figura 121. Ejemplos de imágenes generadas con Data Augmentation – Dataset Perú.....	149
Figura 122. Función features	149
Figura 123. Función BOVW	150
Figura 124. Normalización de los datos	150
Figura 125. Ejemplo de los datos normalizados	151
Figura 126. Selección de parámetros	151

Figura 127. Obtención de los vectores característicos con HOG	152
Figura 128. Asignación del TARGET para los vectores característicos SVM.....	152
Figura 129. Función shuffle para los vectores característicos	152
Figura 130. Función para particionar los datos y aplicar SVM	153
Figura 131. Aplicación de GridSearchCV con SVM.....	153
Figura 132. Cargando el modelo VGG16	154
Figura 133. Agregación de dos capas a la arquitectura de VGG16	154
Figura 134. Código para compilar y entrenar una red	155
Figura 135. Extracción de la penúltima capa de la red	157
Figura 136. Construcción de la tabla de CNN + ML	157
Figura 137. Aplicación de GridSearchCV	157
Figura 138. Gráfica de Accuracy VGG16 (meandros) – Dataset Brasil.....	162
Figura 139. Gráfica de Loss VGG16 (meandros) – Dataset Brasil	162
Figura 140. Gráfica de Accuracy VGG19 (meandros) – Dataset Brasil.....	163
Figura 141. Gráfica de Loss VGG19 (meandros) – Dataset Brasil	163
Figura 142. Gráfica de Accuracy Inception (meandros) – Dataset Brasil	164
Figura 143. Gráfica de Loss Inception (meandros) – Dataset Brasil.....	164
Figura 144. Gráfica de Accuracy ResNet (meandros) – Dataset Brasil	165
Figura 145. Gráfica de Loss ResNet (meandros) – Dataset Brasil	165
Figura 146. Gráfica de Accuracy LeNet (meandros) – Dataset Brasil	166
Figura 147. Gráfica de Loss LeNet (meandros) – Dataset Brasil	166
Figura 148. Gráfica de Accuracy VGG16 (espirales) – Dataset Brasil	167
Figura 149. Gráfica de Loss VGG16 (espirales) – Dataset Brasil.....	168
Figura 150. Gráfica de Accuracy VGG19 (espirales) – Dataset Brasil	168
Figura 151. Gráfica de Loss VGG19 (espirales) – Dataset Brasil.....	169
Figura 152. Gráfica de Accuracy Inception (espirales) – Dataset Brasil.....	169
Figura 153. Gráfica de Loss Inception (espirales) – Dataset Brasil	170
Figura 154. Gráfica de Accuracy ResNet (espirales) – Dataset Brasil	170
Figura 155. Gráfica de Loss ResNet (espirales) – Dataset Brasil.....	171
Figura 156. Gráfica de Accuracy LeNet (espirales) – Dataset Brasil.....	171
Figura 157. Gráfica de Loss LeNet (espirales) – Dataset Brasil.....	172
Figura 158. Ejemplo de GridSearchCV para SVM, RF y KNN.....	172
Figura 159. Gráfica de Accuracy VGG16 (meandros) – Dataset Perú.....	177
Figura 160. Gráfica de Loss VGG16 (meandros) – Dataset Perú.....	178

Figura 161. Gráfica de Accuracy VGG19 (meandros) – Dataset Perú.....	178
Figura 162. Gráfica de Loss VGG16 (meandros) – Dataset Perú.....	179
Figura 163. Gráfica de Accuracy Inception (meandros) – Dataset Perú	179
Figura 164. Gráfica de Loss Inception (meandros) – Dataset Perú	180
Figura 165. Gráfica de Accuracy ResNet (meandros) – Dataset Perú.....	180
Figura 166. Gráfica de Loss ResNet (meandros) – Dataset Perú	181
Figura 167. Gráfica de Accuracy LeNet (meandros) – Dataset Perú	181
Figura 168. Gráfica de Loss LeNet (meandros) – Dataset Perú	182
Figura 169. Gráfica de Accuracy VGG16 (espirales) – Dataset Perú	183
Figura 170. Gráfica de Loss VGG16 (espirales) – Dataset Perú	183
Figura 171. Gráfica de Accuracy VGG19 (espirales) – Dataset Perú	184
Figura 172. Gráfica de Loss VGG19 (espirales) – Dataset Perú	184
Figura 173. Gráfica de Accuracy Inception (espirales) – Dataset Perú	185
Figura 174. Gráfica de Loss Inception (espirales) – Dataset Perú.....	185
Figura 175. Gráfica de Accuracy ResNet (espirales) – Dataset Perú	186
Figura 176. Gráfica de Loss ResNet (espirales) – Dataset Perú	186
Figura 177. Gráfica de Accuracy LeNet (espirales) – Dataset Perú	187
Figura 178. Gráfica de Loss LeNet (espirales) – Dataset Perú.....	187
Figura 179. Arquitectura para el usuario	190
Figura 180. Primera interacción con el asistente	191
Figura 181. Opción Información sobre el Parkinson	192
Figura 182. Opciones de cómo se debe realizar la Prueba Rápida	194
Figura 183. Examen escrito online	195
Figura 184. Ejemplo de resultado sobre la Prueba en Línea – Modelo de Clasificación	196
Figura 185. Recolección de datos personales a través del AV	197
Figura 186. Opción Ayuda Personalizada	198
Figura 187. Código para guardar en la base de datos	199
Figura 188. Estructura tabla EPA base de datos	200
Figura 189. Ejemplo de registro guardado.....	201
Figura 190. Diagrama de flujo del proceso en NodeRed.....	202
Figura 191. Arquitectura Node-RED.....	204
Figura 192. Tab General del Dashboard.....	205
Figura 193. Tab Detalle Respuestas del Dashboard	206
Figura 194. Tab Información Respuestas del Dashboard	207

Figura 195. Arquitectura para el miembro de la Asociación	207
Figura 196. Examen de escritura en vivo.....	208
Figura 197. Plataforma para el analizar las imágenes y realizar la clasificación	208

Índice de Tablas

Tabla 1. Análisis experimental con las diferentes redes convolucionales.....	16
Tabla 2. Elementos bien clasificados en el test considerando las imágenes de los meandros.....	17
Tabla 3. Tabla de elementos bien clasificados en el test considerando las imágenes de los espirales.....	17
Tabla 4. Resumen del mejor rendimiento del clasificador.	21
Tabla 5. Definiciones de IA del punto de vista de diferentes autores.....	47
Tabla 6. Películas con el número de patadas, besos y tipo por cada película.....	52
Tabla 7. Distancias entre cada película y la película sin tipo	53
Tabla 8. Tabla de Población y Muestra	80
Tabla 9. Matriz de variables principales	82
Tabla 10. Actividades etapa de Adquisición.....	86
Tabla 11. Actividades etapa de Preprocesamiento primer dataset.....	89
Tabla 12. Actividades etapa de Preprocesamiento segundo dataset	94
Tabla 13. Actividades etapa de extracción de características	97
Tabla 14. Actividades etapa de modelado y clasificación – Metodología 1.....	104
Tabla 15. Actividades etapa de modelado y clasificación – Metodología 2.....	111
Tabla 16. Actividades etapa de modelado y clasificación – Metodología 3.....	122
Tabla 17. Cronograma de Actividades.....	130
Tabla 18. Presupuesto	132
Tabla 19. Estructura de base de datos.....	133
Tabla 20. Parámetros de las redes para meandros – Dataset Brasil.....	155
Tabla 21. Parámetros de las redes para espirales – Dataset Brasil	156
Tabla 22. Parámetros de las redes para meandros – Dataset Perú	156
Tabla 23. Parámetros de las redes para espirales – Dataset Perú.....	156
Tabla 24. Parámetros para GridSearchCV - SVM.....	158
Tabla 25. Parámetros para GridSearchCV - RF.....	158
Tabla 26. Parámetros para GridSearchCV - KNN.....	158
Tabla 27. Resultados con $k = 10$ – Dataset Brasil	159
Tabla 28. Resultados con $k = 50$ – Dataset Brasil	159
Tabla 29. Resultados con $k = 100$ – Dataset Brasil	160
Tabla 30. Resultados con HOG – Dataset Brasil.....	160
Tabla 31. Resultados con CNN (meandros) – Dataset Brasil.....	161

Tabla 32. Resultados con CNN (Espirales) – Dataset Brasil.....	167
Tabla 33. Resultados con CNN + ML (meandros) – Dataset Brasil.....	173
Tabla 34. Resultados con CNN + ML (espirales) – Dataset Brasil	173
Tabla 35. Resultados con $k = 10$ – Dataset Perú.....	174
Tabla 36. Resultados con $k = 50$ – Dataset Perú.....	175
Tabla 37. Resultados con $k = 100$ – Dataset Perú.....	175
Tabla 38. Resultados con HOG – Dataset Perú	176
Tabla 39. Resultados con CNN (meandros) – Dataset Perú	177
Tabla 40. Resultados con CNN (espirales) – Dataset Perú.....	182
Tabla 41. Resultados con CNN + ML (meandros) – Dataset Perú.....	188
Tabla 42. Resultados con CNN + ML (meandros) – Dataset Perú.....	189
Tabla 43. Flujo de preguntas de síntomas de la EP	193

Resumen

Actualmente, la Enfermedad de Parkinson es la segunda enfermedad neurodegenerativa con más presencia en la población a nivel mundial, afecta directamente la calidad de vida y actividades diarias de una persona. Esta enfermedad está en aumento no solo en el Perú sino en el mundo. Por lo cual, el objetivo de la investigación es implementar un modelo de visión computacional para el pre-diagnóstico de la Enfermedad de Parkinson (EP) a partir de la escritura de una persona. El presente trabajo de investigación es de tipo experimental, explicativo y de enfoque cuantitativo, las fases son las siguientes: Adquisición, Preprocesamiento, Extracción de Características y Modelado y Clasificación. Los resultados de los experimentos alcanzan el nivel del 99% de *Accuracy*, 99% de *Precision*, 99% de *Recall*, 98% de *F1 Score* y 98% de AUC. Se concluye que, se obtuvo y construyó una base de datos que contiene los manuscritos de personas sanas y con EP. Segundo, se utilizaron técnicas de preprocesamiento, las cuales permitieron mejorar la calidad de las imágenes. Tercero, para la construcción del algoritmo, se hizo un procesamiento a las imágenes, se realizaron experimentos con descriptores SIFT, SURF, ORB y HOG para la extracción de características. Cuarto, se utilizó SVM como modelo de clasificación de *Machine Learning (ML)*, además, se utilizaron redes convolucionales con distintas arquitecturas como VGG16, VGG19, Inception, ResNet50 y LeNet y finalmente se utilizaron técnicas que combinan las CNN + ML, con los modelos *SVM*, *RF* y *KNN*.

Palabras clave: Enfermedad de Parkinson, escritura, preprocesamiento de imágenes, visión computacional, redes neuronales convolucionales, prediagnóstico

Abstract

Currently, Parkinson's disease is the second neurodegenerative disease with the most presence in the population worldwide, it directly affects the quality of life and daily activities of a person. This disease is on the rise not only in Peru but in the world. Therefore, the objective of the research is to implement a computer vision model for the pre-diagnosis of Parkinson's disease (PD) from the writing of a person. This research work is experimental, explanatory and quantitative approach, the phases are as follows: Acquisition, Preprocessing, Characteristics Extraction and Modeling and Classification. The results of the experiments reach the level of 99% Accuracy, 99% Precision, 99% Recall, 98% F1 Score and 98% AUC. It is concluded that a database containing the manuscripts of healthy people and with PD was obtained and constructed. Second, preprocessing techniques were used, which allowed to improve the quality of the images. Third, for the construction of the algorithm, images were processed, experiments were carried out with SIFT, SURF, ORB and HOG descriptors for the extraction of characteristics. Fourth, SVM was used as a Machine Learning (ML) classification model, in addition, convolutional networks with different architectures such as VGG16, VGG19, Inception, ResNet50 and LeNet were used and finally techniques that combine CNN + ML, with models SVM, RF and KNN.

Keywords: Parkinson's disease, writing, image preprocessing, computer vision, convolutional neural networks, prediagnosis

Introducción

Actualmente, la tecnología está en avance día a día como se puede observar su uso se está haciendo cada vez más frecuente, ya que desde que uno se levanta, lo primero que hace es revisar el celular, con el cual se entera de todo lo que está sucediendo en tiempo real. Han quedado atrás muchas de las actividades, como por ejemplo las que los padres solían hacer cuando eran niños e incluso las generaciones previas al siglo XXI, las cuales realizaban sin hacer uso de ningún tipo de tecnología. Según una encuesta realizada en 2018 a través de Encuesta Global al Consumidor Móvil, una persona revisa su teléfono aproximadamente 40 veces al día. (Deloitte, 2018).

Por otro lado, cabe resaltar que existen otro tipo de aplicaciones y usos, ya que en pleno siglo XXI, caracterizado por el ya mencionado avance y expansión de la tecnología surgen diferentes ramas de esta como es la inteligencia artificial. Un ejemplo del inicio de la inteligencia artificial es Watson. Según Stroud, Watson es una plataforma de computación cognitiva desarrollada originalmente por IBM para responder preguntas en el programa de preguntas Jeopardy. Watson se basó en el proyecto DeepQA de la empresa, y sus capacidades avanzadas de respuesta a preguntas (QA) ahora están siendo utilizadas por IBM y clientes empresariales en una variedad de aplicaciones. (Stroud, 2018). Al inicio, no fue fácil la construcción de Watson, debido a que el reto era hacer que una máquina entienda como un humano y eso es demasiado complejo, ya que Watson tenía que relacionar todos los conceptos que tenía almacenados y llegar a una conclusión. Por lo tanto, el método que se encontró para que Watson lograra entender fue el *Machine Learning*, cuyo objetivo es desarrollar técnicas que permitan a las máquinas aprender.

Por lo cual, si una máquina puede aprender como lo hace un humano imagine cuántas cosas resultarían más sencillas. Es por ello que surge la necesidad de atender una de las necesidades básicas más importantes de la vida diaria, la salud, puesto que las consecuencias en este campo tienen impacto personal y global.

Hoy en día las computadoras son capaces de observar cómo los humanos. Según la revista Hayo, las computadoras emulan la visión humana, asimismo, los humanos toman decisiones en base a lo que observan, por lo cual las computadoras tienen la misma capacidad. (Hayo, 2017)

Según PARKINSON'S NEWS TODAY, la Enfermedad de Parkinson es la segunda enfermedad neurodegenerativa con más presencia en la población a nivel mundial, afecta

directamente la calidad de vida y actividades diarias de una persona. (PARKINSON’S NEWS TODAY, 2020). Lamentablemente no tiene cura, pero existe la posibilidad de prolongar el efecto de la enfermedad siempre y cuando haya una temprana identificación. Por lo tanto, en el presente trabajo de investigación se da una solución frente a esta problemática creando el diseño de un sistema de pre-diagnóstico de la Enfermedad de Parkinson a través del análisis por visión computacional de la escritura de una persona.

Este trabajo está organizado de la siguiente forma, en el Capítulo 1 denominado Planteamiento del Problema, se detalla la descripción de la realidad problemática, problemas, objetivos, justificación y delimitación del presente trabajo de investigación. En el Capítulo 2 denominado Marco Teórico, se detalla los antecedentes de la investigación, las bases teóricas, el marco conceptual, lo cual sirve como fundamentación teórica para la realización del trabajo de investigación, y la hipótesis. En el Capítulo 3, denominado Metodología, se detalla el diseño, la población y muestra, instrumento de medida, variables, técnicas de recolección de datos, técnicas de procesamiento, cronograma de actividades y presupuesto. En el Capítulo 4, denominado Desarrollo del Experimento, se detalla la adquisición del *dataset*, fases de procesamiento, extracción de características y modelo y clasificación. En el Capítulo 5, denominado Análisis y Discusión de Resultados, se detalla los resultados de los experimentos realizados. Finalmente, en el Capítulo 6, denominado Conclusiones y Recomendaciones, se detalla las conclusiones obtenidas de la realización del presente trabajo de investigación y las recomendaciones a tomar.

Capítulo I: Planteamiento del Problema

En el presente capítulo se habla sobre la realidad problemática, el planteamiento del problema, los objetivos, la justificación y la delimitación, los cuales sirven para poder indicar de qué trata esta investigación, en qué campo se va a estudiar, cuáles son los problemas para solucionar y los objetivos a cumplir. Todo esto será más especificado en la justificación, la cual indica a quién se beneficia con la presente investigación y la delimitación, la cual indica en qué campo se va a aplicar.

1.1 Descripción de la Realidad Problemática

En el presente año 2020, se está atravesando una pandemia debido a la COVID-19, según la Organización Mundial de la Salud (OMS), la COVID-19 es una enfermedad que afecta a muchos países en el mundo y es una enfermedad infecciosa causada por el coronavirus, la cual desencadena infecciones respiratorias en los humanos y en algunos casos causa la muerte. (OMS, 2020). Es por ello, que ante los efectos de las enfermedades surge la necesidad de prevenirla con el fin de evitar el sufrimiento y muertes en las personas.

A nivel mundial la Inteligencia Artificial está siendo utilizada para ayudar y prevenir la COVID-19. Según el medio BBC, en China se han construido robots que son capaces de llevar productos entre personas, desinfectar calles y hospitales. Asimismo, con ayuda de la Inteligencia Artificial, se han desarrollado sistemas que permiten diagnosticar la COVID-19 con una precisión del 96%. Finalmente, se está utilizando la visión computacional para poder controlar la temperatura de una persona e incluso identificarla, sin importar que esté utilizando mascarilla. (BBC, 2020)

En Perú también se está haciendo uso de la Inteligencia Artificial contra la COVID-19, es así como se están construyendo sistemas inteligentes que ayuden a prever y diagnosticar dicha enfermedad. Un ejemplo es el desarrollo de la aplicación “Perú en tus manos”, que permite a los ciudadanos visualizar las zonas que tienen mayor probabilidad de contagio, estas zonas se actualizan en base a los nuevos casos positivos de COVID-19 con el fin de ser la primera alerta para los ciudadanos al salir de sus casas. (PCM, 2020). Asimismo, se implementó la Encuesta Nacional COVID-19, la cual se estima que llegue a 20 millones de peruanos, esta encuesta llega a todo tipo de celular, debido a que utiliza la tecnología USSD cuyo objetivo es conocer las zonas de potencial riesgo e identificar posibles casos nuevos de COVID-19. (PCM, 2020). Por lo cual, se puede identificar que la construcción y el uso de sistemas inteligentes ayuda a prevenir y combatir las enfermedades. De ese modo, en el

presente trabajo de investigación se plantea utilizar sistemas inteligentes para prevenir una de las enfermedades más comunes, la enfermedad de Parkinson.

Según la OMS, en el año 2018 había 7 millones de personas diagnosticadas con la enfermedad de Parkinson, siendo afectadas en su mayoría las personas mayores a 60 años. Además, la OMS ha pronosticado que por el año 2030 la cantidad de personas afectadas por esta enfermedad serían más de 12 millones (Figura 1). No existe una causa en específico para justificar la presencia de la enfermedad de Parkinson, esta no distingue ni por sexo, raza, cultura, entre otros, se puede presentar en cualquier persona y afecta principalmente el movimiento en actividades diarias como caminar, levantarse de una silla, temblor en reposo, falta de equilibrio, micrografía, lentitud de movimientos entre otros. Cabe resaltar, que no afecta del mismo modo a todas las personas, algunas presentan otro tipo de manifestaciones como dolor, ansiedad o depresión. (OMS, 2018).

En una actualización más reciente, según PARKINSON'S NEWS TODAY aproximadamente más de 10 millones de personas en el mundo presentan la enfermedad de Parkinson, siendo esta la segunda enfermedad neurodegenerativa más común después de la enfermedad del Alzheimer. (PARKINSON'S NEWS TODAY, 2020).

Por lo tanto, se puede identificar que el número de personas que presentan esta enfermedad está aumentando de manera progresiva, por lo cual surge la necesidad de preverla, ya que esta enfermedad no es mortal.

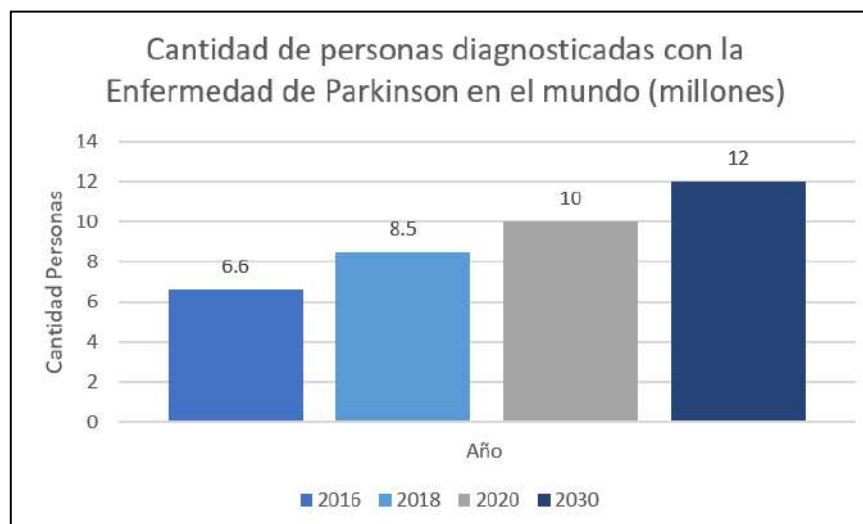


Figura 1. Cantidad de personas diagnosticadas con la Enfermedad de Parkinson en el mundo según la OMS y PARKINSON'S NEWS TODAY.

Fuente: Elaboración propia

En el Perú la realidad no es muy distante, lo que actualmente es una enfermedad mundial también tienen un impacto en los peruanos. Según el Instituto Nacional de Ciencias Neurológicas (INCN), en el Perú no se sabe con exactitud cuántos pacientes presentan la enfermedad de Parkinson, pero aproximadamente son 30 mil personas. Además, cada año se presentan 3000 nuevos casos y el diagnóstico en los jóvenes está en aumento (Figura 2). (INCN, 2019). Se puede observar que estas cifras no son tan elevadas dado que, si hacemos una comparación, existen otro tipo de enfermedades desde la epilepsia hasta la enfermedad de Alzheimer que están presentes en mayor cantidad en la población peruana y mundial. Sin embargo, es importante resaltar que las cifras están en aumento como se mencionó anteriormente, por lo cual es mejor tomar medidas de precaución lo más antes posible.

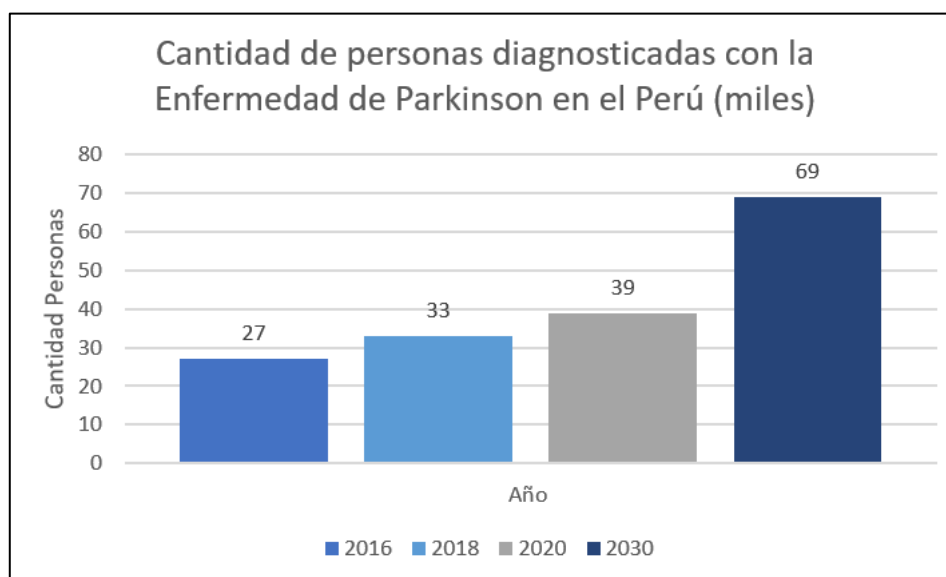


Figura 2. Cantidad de personas diagnosticadas con la Enfermedad de Parkinson en el Perú según Instituto Nacional de Ciencias Neurológicas (INCN).

Fuente: Elaboración propia

El National Insitute of Neurological Disorders and Stroke (NIH) indica que no existe una prueba certera para detectar la enfermedad. Además, en el momento en el que una persona está envejeciendo, generalmente presentan diferentes cambios en su cuerpo, por lo cual pueden pensar que dichos cambios son producto de la edad, pero en algunos casos se pueden estar evidenciando los síntomas de la enfermedad de Parkinson y estos pueden pasar desapercibidos por las personas. Sin embargo, uno de los principales síntomas es el temblor, movimiento rítmico, que puede aparecer en las manos, principalmente en el dedo pulgar e índice. (NIH, 2016). Las personas sean zurdas o diestras al momento de escribir hacen uso de los dedos pulgar e índice, por lo cual se puede buscar patrones a partir de la escritura de una persona, ya

que una persona con la enfermedad de Parkinson tiende a presentar movimiento anormal en la mano, a diferencia de una persona sana, al momento de escribir.

Las actividades diarias que las personas realizan son importantes, es más se involucra a otros individuos al realizarlas, ya sean sus familiares, amigos, profesores, compañeros de trabajo, entre otros. Sin embargo, qué sucede cuando se tiene que cambiar las rutinas o dejar de hacer lo que les gusta. Muchas personas en Perú y el mundo tienen que cambiar su modo de vivir al darse cuenta de que presentan cualquier tipo de enfermedad. Según NIH, la vida diaria de una persona, la cual presente la enfermedad de Parkinson puede estar afectada al momento de hacer las actividades domésticas, ir a trabajar, socializar con otras personas, entre otros, dado que conforme avance la enfermedad el cuerpo tiende a dejar de realizar actividades motoras. Sin embargo, el primer paso es que uno mismo acepte que tiene la enfermedad de Parkinson y es muy importante que el entorno social de la persona que padece dicha enfermedad la apoye. Además, actualmente no existe una cura para la enfermedad de Parkinson, no obstante, existen diferentes tipos de terapia que pueden ayudar a que el proceso de Parkinson aparezca más lentamente como medicamentos, cirugías, terapias de apoyo, ejercicio, dieta. (NIH, 2016).

Ante estas alternativas de terapia, existe la posibilidad de prolongar el efecto de la enfermedad de Parkinson siempre y cuando haya una temprana identificación de dicha enfermedad. Por lo cual, es recomendable realizarse análisis de detección una o dos veces al año para poder preverla y prolongarla con el fin de llevar una vida diaria común de acuerdo a cómo una persona se desarrolla siempre, asimismo, una detección temprana puede ayudar a sobrellevar mejor la enfermedad. Cabe resaltar, que la mencionada recomendación debería aplicarse en todas las detecciones de enfermedades, con el fin de evitar contraer una enfermedad, gastar en terapias, medicamentos, operaciones, entre otros e inclusive en algunos casos más críticos se podría evitar la muerte. Por ende, surge la disyuntiva sobre qué se puede hacer para detectar y prever la enfermedad de Parkinson. Actualmente, las ciencias de la computación están avanzando de manera significativa y se pronostica que van a seguir siendo así en los próximos años. Una de las ramas de la ciencia de la computación es la Inteligencia Artificial (IA), la cual es “el estudio de saber cómo hacer que las computadoras hagan cosas que hasta el momento las personas hacen mejor”. (Rich and Knight, 1991). Además, dentro de la IA se encuentra la Visión Computacional (VC) con la cual se puede hacer uso de una de sus principales aplicaciones como el reconocimiento óptico de caracteres (Figura 3), el cual se puede aplicar para el reconocimiento de escritura a mano, de placas, de códigos postales.

(Richard Szeliski, 2010). Por lo tanto, se puede hacer uso de estos métodos, los cuales una computadora puede hacer de manera más eficiente que un humano, para encontrar soluciones efectivas frente un problema.

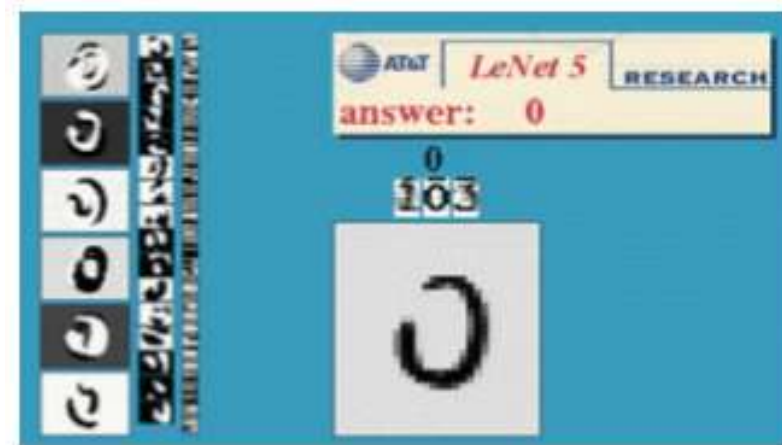


Figura 3. Reconocimiento óptico de caracteres

Fuente: Rich & Knight (1991). *An Introduction to Artificial Intelligence*. (second edition). (p. 6)

La predicción de si una persona presenta indicios o tiene la enfermedad de Parkinson es posible a través de distintos modelos de *Machine Learning*, siendo aquella que se encarga de aprender con data mediante uso de técnicas estadísticas (Ethem Alpaydin, 2016), los cuales van a ayudar a clasificar y determinar el resultado de la variable a predecir según sea el caso. Con la finalidad de detectar tempranamente o prever si una persona tiene la enfermedad de Parkinson, se planeó la creación de un sistema de reconocimiento de escritura de una persona, para su posterior análisis mediante visión computacional con el objetivo de clasificar las imágenes de dicha escritura en dos grupos, los cuales serían si la persona presenta o no presenta la enfermedad de Parkinson. Por lo cual, se pretende que una persona posea conocimiento previo acerca de si es propenso a tener la enfermedad y de ese modo pueda hacerse una consulta médica temprana.

Este tipo de sistemas podrían ayudar a que una persona pueda hacerse la prueba de una manera sencilla y práctica en tiempo real y en línea. Finalmente, lo más importante que se puede lograr sería reducir los cambios significativos que la persona tenga que hacer en las actividades que realiza en la vida diaria, ya que como se mencionó anteriormente, la calidad de vida empieza a bajar conforme la enfermedad de Parkinson aumenta.

1.2 Formulación del Problema

Para la identificación del problema general y de los problemas específicos se hizo uso de la matriz de contingencia, la cual se puede visualizar en el Anexo 1. Asimismo, los problemas han sido identificados en base al proceso de captación de datos que se va a dar a lo largo de la investigación y en base a una realidad actual que es la enfermedad de Parkinson.

1.2.1 problema general

PP: ¿Es posible utilizar técnicas de visión computacional para el prediagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona?

1.2.2 problemas específicos

PE1: ¿Existen bases de datos disponibles de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson?

PE2: ¿Qué técnicas de procesamiento de imágenes se pueden aplicar a imágenes de manuscritos para poder aumentar la calidad de la imagen?

PE3: ¿Qué técnicas de visión computacional existen para transformar a vectores característicos descriptores como color, forma o textura de las imágenes?

PE4: ¿Qué tipos de técnicas de *machine learning* y *deep learning* son posibles de aplicar para poder clasificar si la imagen del manuscrito pertenece a una persona con prediagnóstico de la enfermedad de Parkinson?

PE5: ¿Qué métricas permiten evaluar la eficiencia de la clasificación si la imagen del manuscrito pertenece a una persona con prediagnóstico de la enfermedad de Parkinson?

1.3 Objetivos de la Investigación

Para la identificación del objetivo general y de los objetivos específicos se hizo uso de la técnica de la matriz de contingencia, la cual se puede visualizar en los anexos. Asimismo, los objetivos han sido identificados en base a los problemas previamente identificados con el fin de dar una solución.

1.3.1 objetivo general

OP: Implementar un modelo de visión computacional para el prediagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona.

1.3.2 objetivos específicos

OE1: Identificar y crear bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson.

OE2: Implementar técnicas de procesamiento de imágenes a imágenes de manuscritos para poder aumentar la calidad de la imagen.

OE3: Implementar técnicas de visión computacional para obtener vectores característicos que describan el color, forma o textura de las imágenes.

OE4: Utilizar técnicas de *machine learning* y *deep learning* para poder clasificar si la imagen del manuscrito pertenece a una persona con prediagnóstico positivo de la enfermedad de Parkinson.

OE5: Utilizar métricas que permitan evaluar la eficiencia de la clasificación si la imagen del manuscrito pertenece a una persona con prediagnóstico de la enfermedad de Parkinson.

1.4 Justificación de la Investigación

1.4.1 teórica

Esta investigación se realiza con el propósito de determinar si se pueden utilizar técnicas de visión computacional para poder realizar el procesamiento de imágenes a fin de aumentar la calidad, obtener puntos característicos y patrones de las mismas. Además, utilizando algoritmos de *Machine Learning* y *Deep Learning* crear modelos de clasificación para determinar si se puede pre-diagnosticar la enfermedad de Parkinson a través de la escritura de una persona con el objetivo de mejorar la calidad de vida de las personas, ya que actualmente no existe una prueba general que lo determine y uno de los principales síntomas de la enfermedad de Parkinson es el temblor en las manos.

Existen investigaciones donde se comprueba que la visión computacional e inteligencia artificial están avanzando significativamente. En los últimos años, el reconocimiento de escritura y las técnicas de preprocesamiento son más investigados, debido a los nuevos modelos de *Deep Learning* y los recursos computacionales que se tienen. El reconocimiento de escritura y número cada vez es más preciso gracias a este tipo de técnicas. (Russel y Norvig, 2010)

Por último, en el Perú y en el mundo este tipo de estudios no son muy comunes. Por lo cual, existe la oportunidad de dar una solución que ayude a la literatura de nuevas investigaciones y a muchas personas que presente la mencionada enfermedad.

1.4.2 práctica

Al término de esta investigación, las personas podrán hacer uso del sistema de pre-diagnóstico de la enfermedad de Parkinson con el fin de descartar la aparición de dicha enfermedad de ser la prueba negativa. Sin embargo, de ser positiva y con un porcentaje elevado de certeza, dichas personas deberían realizarse un chequeo neurológico para recibir ayuda profesional y empezar con un tratamiento que les permita desarrollarse de manera normal. Los beneficiarios serán todas las personas que sientan que presentan alguno de los síntomas de la enfermedad de Parkinson, ya que uno de los principales es el temblor en la mano creando un efecto anómalo al momento de escribir. Esta investigación va a servir para evitar el avance de la enfermedad de Parkinson, principalmente en las personas mayores a 60 años, las cuales son más propensas a presentarla.

La presente investigación demostrará que un sistema de visión computacional puede predecir, mediante las imágenes de escritura de las personas, si una persona presenta la enfermedad de Parkinson o no, lo cual puede cambiar la calidad de vida de todas las personas involucradas no solo de aquellas que la poseen, sino de familiares, amigos, compañeros de trabajo, entre otros, ya que si esta enfermedad es tratada con anticipación se puede reducir los efectos de esta.

1.4.3 metodológica

La implementación de un sistema de pre-diagnóstico de la enfermedad de Parkinson ayudará a que se pueda mejorar la calidad de vida de una persona, ya que si esta enfermedad es detectada a tiempo se puede dar un tratamiento adecuado que podría ayudar a prolongar los efectos de dicha enfermedad. Cabe resaltar que, si bien esta enfermedad no es mortal, si no es detectada a tiempo puede generar en la persona complicaciones que le impidan realizar sus actividades diarias, lo cual crea un desbalance entre ella y su entorno. Por consiguiente, el sistema a implementar deberá evaluar los resultados que genere para poder así brindar información para una correcta toma de decisiones en el futuro respecto a una consulta neurológica.

Para definir el pre-diagnóstico de la enfermedad de Parkinson en una persona se utilizará una base de datos llamada *HandPD* y *New HandPD* las cuales están divididas en dos grupos de personas: las personas que presentan y las que no presentan la enfermedad de Parkinson, dichas imágenes fueron recolectadas por la Facultad de Medicina Botucatu, Universidad Estatal de São Paulo, Brasil. Asimismo, se creará una nueva base de datos que

contenga imágenes de los dos grupos, es decir personas que presentan la EP y personas que no. Las imágenes se encuentran en formato “jpg”, son aquellas que van a ser analizadas mediante visión computacional para detectar patrones anómalos y así poder realizar un correcto pre-diagnóstico. Asimismo, se hará uso de técnicas de *Machine Learning* y *Deep Learning* para una correcta clasificación y predicción.

1.5 Delimitación del Estudio

1.5.1 espacial

El presente trabajo de investigación se realizará con la información de unas personas, recolectados por la Facultad de Medicina Botucatu, Universidad Estatal de São Paulo, Brasil, la cual consiste en unas imágenes de escritura realizadas por personas las cuales se dividen en dos grupos: las personas que presentan y las que no presentan la enfermedad de Parkinson. Los datos, son dichas imágenes que se encuentran en formato “jpg”, son extraídos de una base de datos llamada *HandPD* y *New HandPD* consisten en formas de espirales y meandros. Además, de la utilización de una nueva base de datos con imágenes de manuscritos de personas con EP y sin EP, estas imágenes, del mismo modo, que la base de datos de Brasil, consisten en formas espirales y meandros. Asimismo, en una sección se encuentra la escritura de las mencionadas personas. Por último, se va a hacer uso de *papers* que contengan resultados previos de investigaciones relacionadas a la predicción de la enfermedad de Parkinson para poder hacer una comparación de la efectividad del sistema a desarrollarse y se hará uso de libros y documentación externa.

1.5.2 temporal

La presente investigación analizará un conjunto de datos de escritura obtenidos de la ya mencionada base de datos *HandPD* y *New HandPD*, en el año 2016 y de la nueva base de datos de escritura del año 2020. Estas bases de datos han sido realizadas, dado que se quería analizar el comportamiento de la escritura en las personas, con el fin de detectar anomalías. Han pasado cuatro años y en la presente investigación tenemos por objetivo verificar y mejorar los resultados obtenidos con los datos de Brasil. Asimismo, obtener nuevos algoritmos y resultados que validen que las mismas técnicas pueden ser aplicadas en diferentes contextos y con una nueva base de datos.

1.5.3 conceptual

Esta investigación se centrará en el análisis de imágenes de escritura, en la clasificación y predicción de que una persona presente la enfermedad de Parkinson o no. Para lo cual se

necesitará herramientas de Inteligencia Artificial, las cuales son Visión Computacional y *Machine Learning*. Existen diferentes síntomas que pueden indicar que una persona presenta la enfermedad de Parkinson, pero en este caso utilizaré uno de los principales el cual es el temblor anómalo que se presenta en la mano. El campo en el que se aplicará esta solución es en Medicina y se busca poder dar una contribución de manera efectiva con el fin de mejorar la calidad de vida de las personas que presentan la mencionada enfermedad.

Capítulo II: Marco Teórico

En el presente capítulo se va a dar paso a la explicación de aquellos antecedentes, bases y conceptos teóricos, los cuales van a ser de ayuda para poder entender mejor el problema y las definiciones necesarias, ya que para una buena investigación y resultados es conveniente tener una idea previa para lo que se va a exponer. Asimismo, en este capítulo se van a evaluar resultados previos de autores que ya han investigado acerca del tema.

2.1 Antecedentes de la Investigación

A continuación, se va a presentar algunos antecedentes encontrados para poder entender cómo es que se ha tratado en otros casos el tema de investigación elegido, los objetivos, conclusiones y las soluciones a las cuales se ha llegado. Cabe resaltar que, hasta la fecha no se tiene conocimiento que exista una investigación de este tipo a nivel nacional.

En la investigación publicada en 2016 titulada *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics* cuyos autores son R. Clayton, T. Slike, Hook Christian, H. Gustavo & P. Joao. Se sostiene que la enfermedad de Parkinson es una de las más difíciles de detectar en las primeras etapas. En su investigación ellos plantean hacer una detección automática de la enfermedad de Parkinson a través de una red neuronal convolucional (CNN), quien se encarga de procesar diversos tipos de entrada como en este caso imágenes, las cuales fueron tratadas y cuyos pixeles fueron convertidos a un vector de características para posteriormente encontrar patrones en los cuales se puedan detectar anomalías y así crear una diferencia entre aquellas imágenes que representan a una persona sana y aquellas que representan a una persona propensa o con la enfermedad de Parkinson.

La metodología usada consistió en tres etapas. Primero, el diseño de la base de datos, lo que se hizo fue captar datos mediante unas hojas de escritura (Figura 4) que se le daba a pacientes sanos y enfermos, ya que cuando una persona presenta la enfermedad de Parkinson uno de los síntomas es un temblor anómalo en la mano, así que se obtuvieron esas imágenes a través de una pluma especial y las señales generadas a través de los movimientos (Figura 5),

cabe indicar que las pruebas fueron extraídas de la Facultad de Medicina Botucatu, Universidad Estatal de Sao Paulo, Brasil.

22/11/2010

Field study: Unesp 2010
University of Applied Sciences
Regensburg
Biometric Smart Pen Project
Universidade Estadual Paulista
Faculdade de Medicina (FMB),
Botucatu

Distúrbio e Hiperkinesia Medicamentosa
Tomou medicamento às 14h, agora é 15h30

Idade: 56 Mão dominante: ☒ direita () esquerda



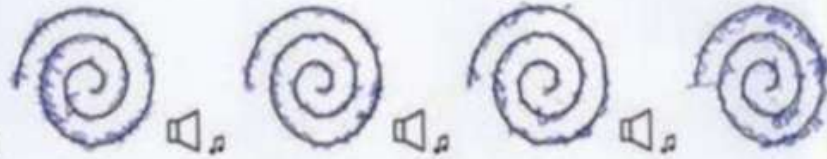
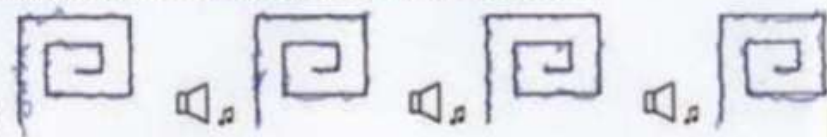


a, b	Desenhar círculo 12 vezes no mesmo lugar sem parar. 	Desenhar círculo no ar 12 vezes no mesmo lugar sem parar. 
c	Desenhar espiral após sinal sonoro, de dentro para fora. 	
d	Desenhar meander após sinal sonoro, de dentro para fora. 	
e	Diadococinesia: Mão direita 20 segundos. 	
f	Diadococinesia: Mão esquerda 20 segundos. 	

Figura 4. Modelo de prueba escrita la cual fue entregada a los pacientes

Fuente: Clayton, R., Slike, T., Hook, C., Gustavo, H. & Joao, P. (2016). *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics*. (p. 3)

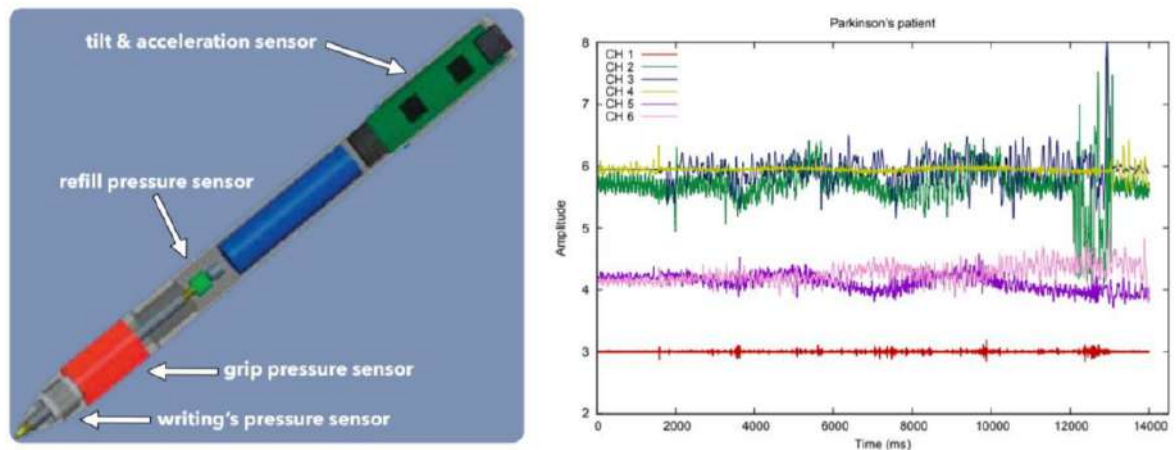


Figura 5. Pluma especial que capta a través de señales los movimientos de la mano y señales captadas por la pluma

Fuente: Clayton et al., (2016). *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics*. (p. 3)

En la segunda etapa, se realizó un tratamiento respectivo a las imágenes con el fin de que puedan ser extraídas de manera correcta más adelante. Los tratamientos que se hicieron fueron la escalación, normalización de los valores de las imágenes y transformarlas a escala de grises.

Finalmente, en la tercera etapa, se realizaron las fases experimentales. La clasificación se hizo mediante redes neuronales convolucionales, *Optimum Path Forest* (OPF). Los *datasets* fueron divididos en dos, uno con los meandros y el otro con los espirales. Ambos estuvieron compuestos por 308 imágenes. Para la fase del modelado los *datasets* fueron divididos en *train* y *test* utilizando dos proporciones distintas, la primera fue de 50% *train* y 50% *test* y en la segunda 75% *train* y 25% *test*. Además, se utilizó técnicas como Validación Cruzada con 20 iteraciones. Diferentes arquitecturas de CNN fueron usadas para proveer un análisis experimental profundo, entre ellas *ImageNet*, CIFAR-10 y *LeNet*.

Tabla 1.

Análisis experimental con las diferentes redes convolucionales.

	ImageNet	CIFAR-10	LeNet
Capas de convolución	5	3	2
Agrupación de capas	5	3	2
Capas de Normalización	2		

Capas de ReLu (Función de Activación)	5	3	1
Capas internas del producto	2	2	2
Capa de Abandono	2		
Capa de Pérdida Softmax	1	1	1
Capa de Rendimiento	1	1	1

Fuente: Clayton et al., (2016). *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics*. (p. 3)

Los resultados presentados con las imágenes de los espirales se pueden resumir en que se obtuvo un 82% de precisión en la tasa de clasificación con CNN y con OPF lograron alrededor de 92%.

Tabla 2.

Elementos bien clasificados en el test considerando las imágenes de los meandros

	50% / 50% (Train / Test)				75% / 25% (Train / Test)			
	64 x 64		128 x128		64 x 64		128 x128	
	Control	PD	Control	PD	Control	PD	Control	PD
ImageNet	74.29%	90.58%	76.31%	87.90%	72.86%	89.55%	76.19%	91.25
CIFAR-10	15.71%	71.92%	10.00%	65.00%	33.33%	82.14%	10.95%	84.20%
LeNet	00.00%	35.00%	00.00%	55.00%	00.00%	00.00%	00.00%	50.00%
OPF	61.91%	86.61%	52.38%	85.71%	61.91%	92.86%	52.38%	92.86%

Fuente: Clayton et al., (2016). *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics*. (p. 5)

En los resultados presentados con las imágenes de los espirales se obtuvieron un 98% de precisión en la tasa de clasificación con CNN y con OPF lograron alrededor de 71% de precisión.

Tabla 3.

Tabla de elementos bien clasificados en el test considerando las imágenes de los espirales

	50% / 50% (Train / Test)				75% / 25% (Train / Test)			
	64 x 64		128 x128		64 x 64		128 x128	
	Control	PD	Control	PD	Control	PD	Control	PD
ImageNet	58.10%	86.03%	56.19%	85.76%	59.52%	87.95%	55.48%	85.80%
CIFAR-10	29.76%	92.77%	06.31%	98.53%	57.38%	86.16%	14.76%	91.79%

LeNet	00.00%	75.00%	00.00%	55.00%	00.00%	60.00%	00.00%	55.00%
OPF	66.67%	90.18%	61.91%	87.50%	71.43%	87.50%	66.67%	82.14%

Fuente: Clayton et al., (2016). *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics*. (p. 6)

Como conclusiones de esta investigación, se puede resumir en que han usado imágenes que fueron obtenidas a través de sensores. Mediante el uso de técnicas supervisadas como CNN se hicieron experimentos para entrenar el modelo y así se pudo lograr tener una buena precisión, por lo cual, en el artículo concluyen diciendo que cumplen con el objetivo de llegar a predecir cuando una persona presenta la enfermedad de Parkinson.

En el artículo publicado en 2018, titulado *Detection of Parkinson's Disease Through Static Analysis of Handwriting and Character Recognition*, cuyos autores son Kalana, R., Sunny L. Se sostiene que uno de los principales síntomas de la enfermedad de Parkinson se presenta en la micrografía, la cual tiene que ver con la escritura, por lo tanto, en esa investigación pretenden a través de las escrituras encontrar cambios relevantes en su morfología y demostrar que el reconocimiento de estos cambios es menos caro y se puede realizar dentro o fuera de una clínica.

La metodología usada consistió en tres etapas. Primero, el *dataset* utilizado en esta investigación fueron las muestras de escritura tanto de personas que presentan la enfermedad de Parkinson como de personas sanas. Segundo, se empleó la técnica de procesamiento de imágenes en las cuáles se realizaron eliminación de ruido, segmentación, recorte, entre otros. Tercero, se empleó la extracción de características, las cuales consisten en encontrar patrones que sean relevantes a la hora de identificar alguna anomalía. En este caso, utilizaron la técnica de aprendizaje supervisado *Support Vector Machine* (SVM), el cual construye un modelo que asigna nuevos datos en una categoría u otra de modo que se trazan puntos en el espacio y de acuerdo con la clasificación estos puntos se encuentran en un lado del margen o en el otro lado. Cuando los datos no están etiquetados, se hace uso del aprendizaje no supervisado, el cual realiza una agrupación de los datos mediante clúster vectores de soporte (SVM). El reconocimiento de caracteres se realizó mediante el algoritmo de detección de borde de *Canny* y el histograma de características. El algoritmo de detección de borde de *Canny* se puede dividir en 4 pasos diferentes:

- Aplicar el filtro gaussiano para suavizar la imagen y eliminar el ruido.

- Encontrar las gradientes de intensidad de la imagen, aplicar una supresión no máxima para eliminar la respuesta espuria a la detección de bordes.
- Aplicar doble umbral para determinar posibles aristas.
- Rastrear borde por histéresis: finalice la detección de bordes al suprimir todos los otros bordes que son débiles y no están conectados a bordes fuertes.

En la Figura 6 se va a describir cómo es la Metodología por fases:

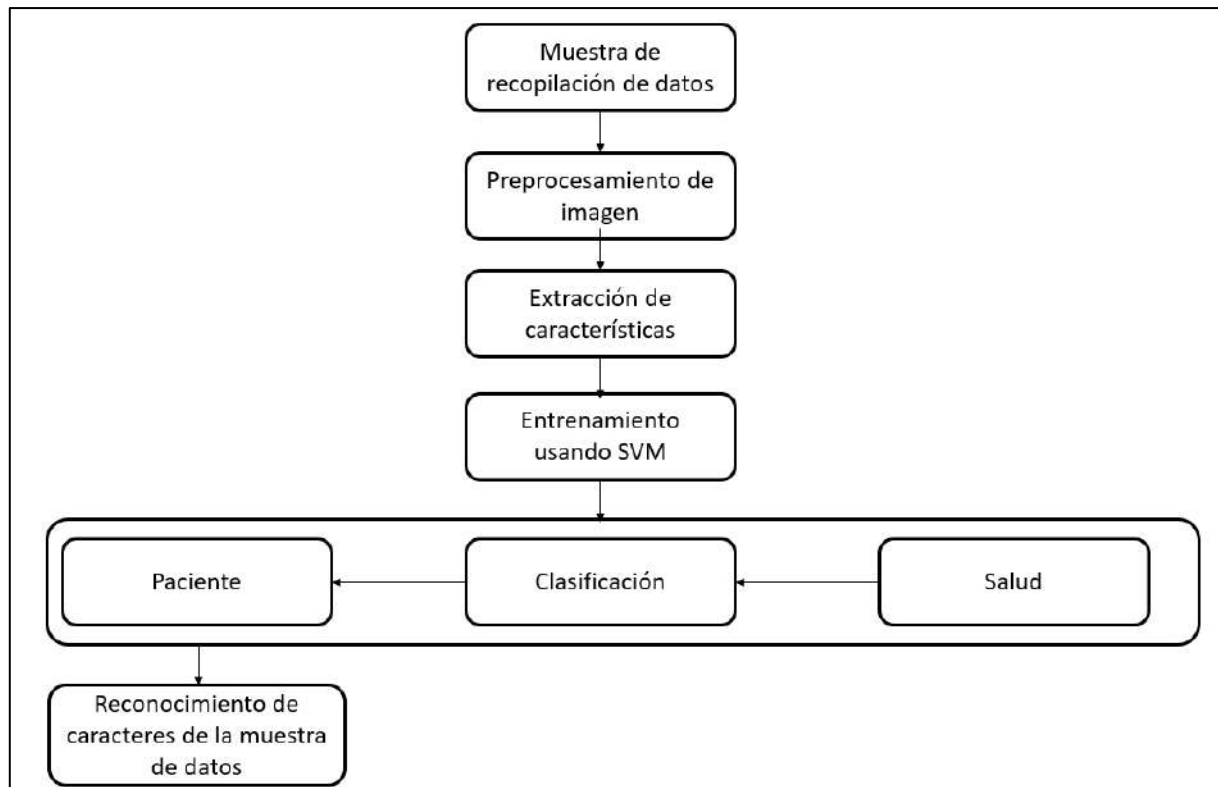


Figura 6. Resumen de la metodología usada en el presente artículo.

Fuente: Kalana, R. & Sunny L. (2018). *Detection of Parkinson's Disease Through Static Analysis of Handwriting and Character Recognition*. (p. 3)

Finalmente, con las imágenes óptimas después de todo el procesamiento está listas para entrar en el modelo SVM, el cual clasificó si la persona presenta o no la enfermedad de Parkinson. Concluyeron que este método es eficiente, ya que actualmente no existe un método oficial para poder detectar la enfermedad de Parkinson y mucho menos fuera de una clínica. Además, este método es menos costoso y el sistema puede aumentar su precisión con más muestras.

En el artículo publicado en 2014, titulado *Diagnosing Parkinson's Disease From Gait*, cuyos autores son Chang, D., Alban, M., Hsu, K. Se sostiene que no hay ninguna prueba exacta

para detectar la enfermedad de Parkinson, la cual afecta a más de un millón de personas. Por lo cual en el siguiente artículo se presentó un clasificador que diagnosticará el Parkinson a partir de mediciones cuantitativas. Se hizo uso de *Support Vector Machine* (SVM) para la clasificación. Asimismo, cabe resaltar que en dicho estudio los datos espacio-temporales se recopilaban utilizando sensores infrarrojos conectados a las caderas y piernas de las personas, mientras que los datos cinéticos fueron recopilados mediante sensores ubicados en los pies de la persona. Dado, que el primero era más complicado de analizar se tomó la decisión de predecir únicamente con los datos cinéticos.

La metodología consistió en 4 etapas. Primero, el dataset fue obtenido del Laboratorio for Gait & Neurodynamics, Movement Disorders Unit of the Tel Aviv Sourasky Medical Center. La data presenta 279 registros de 93 pacientes que presentan la enfermedad de Parkinson.

Segundo, para la extracción de características se tomaron en cuenta aquellos síntomas como la capacidad para mantener un ritmo de marcha estable. Seguidamente, el proceso consistió en extraer los datos cuando la persona está subiendo un escalón durante el cual el pie entra en contacto con el suelo. Luego, se calcula la media y la varianza para cada característica y finalmente se concatena una a otra para crear el vector de características para cada persona.

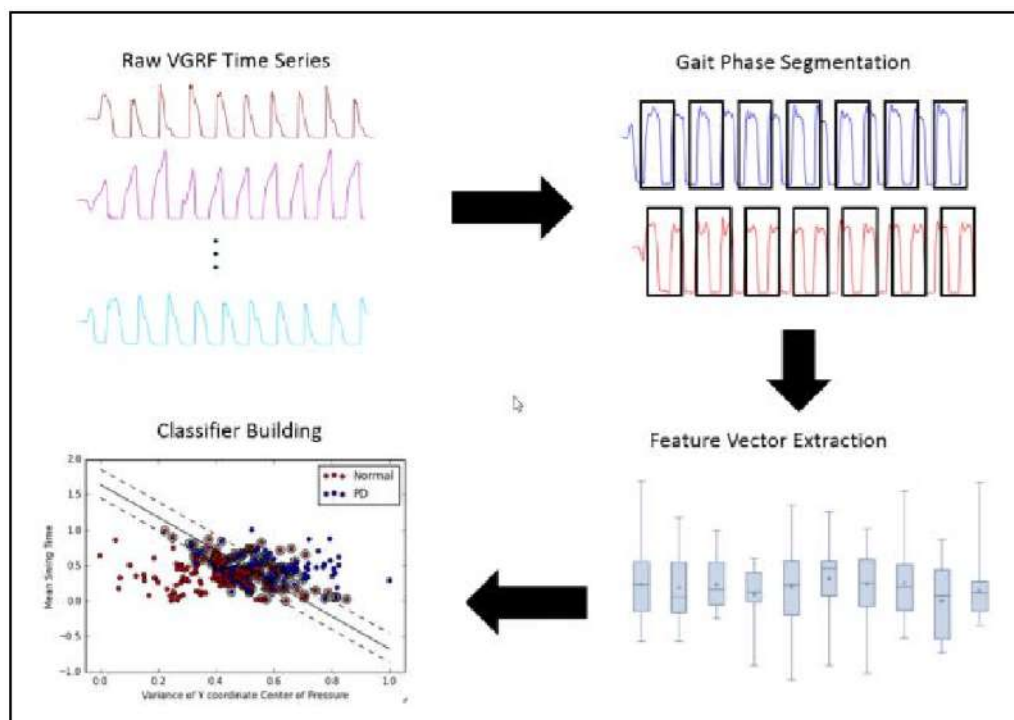


Figura 7. Resumen de la extracción de características.

Fuente: Chang, D., Alban, M. & Hsu, K. (2014). *Diagnosing Parkinson's Disease From Gait*. (p. 1)

Tercero, las características extraídas tienen en cuenta la fuerza, la cual varía en función al peso de la persona. Además, se toman en cuenta los tiempos de giro y la postura, el tiempo de balanceo, el cual es el tiempo desde que el pie se levanta del piso hasta que vuelve a caer al piso. Cuarto, los datos obtenidos pasan por un proceso de normalización para que estos se encuentren en la misma escala y finalmente, se aplican los modelos SVM para la clasificación.

Como resultados, el modelo *Baseline PD Classifier* obtuvo una *Precision* de 70.6%, el cual salió más elevado es el SVM de kernel lineal.

Tabla 4.

Resumen del mejor rendimiento del clasificador.

	AUC	Accuracy	Precision	Recall
Baseline PD Classifier	56.8%	60.2%	70.6%	70.6%
Baseline Severity Classifier	51.5%	49.8%	30.3%	55.0%

Fuente: Chang et al., (2014) *Diagnosing Parkinson's Disease From Gain*. (p. 3)

En el artículo publicado en 2015, titulado *A Step Towards the Automated Diagnosis of Parkinson's Disease: Analyzing Handwriting Movements*, cuyos autores son Clayton, R., Pereira R., Slike T., Hook C. & H. Weber. Se sostiene que, actualmente la Enfermedad de Parkinson es considerada una enfermedad degenerativa y que uno de sus principales efectos es el temblor que se presenta generalmente en las manos y se puede evidenciar cuando una persona escribe. Asimismo, explican que la Micrografía es altamente utilizado para poder realizar un diagnóstico hacia un paciente, y es básicamente un examen escrito. Adicionalmente, otra de las formas para poder realizar un diagnóstico es mediante las resonancias magnéticas. Sin embargo, este método puede ser muy costoso, por lo que los autores proponen poder utilizar sistemas de visión computacional para poder realizar una identificación de patrones en la escritura de los pacientes.

La metodología se dividió en 2 partes. Primero, se enfocaron en conseguir un dataset y los autores mencionan que utilizaron uno proveniente de la Facultad de Medicina de la Universidad de Sao Paulo en Brasil. El *dataset*, está compuesto por unas imágenes las cuales consisten en dibujos por parte del paciente en una prueba. Segundo, se procedió a realizar la extracción de características de las imágenes aplicando técnicas de visión computacional. Lo

que realizaron los autores fue aplicar procesamiento de imágenes y extracción de características, dado que cada imagen tenía un trazo real de la prueba y un segundo trazo de las personas que realizaron la prueba. En la Figura 8 se puede observar cómo es que aplicaron las técnicas ST (*Espiral Template*) y HT (*Handwritten Template*) para poder hacer la separación entre el trazo real y la plantilla.

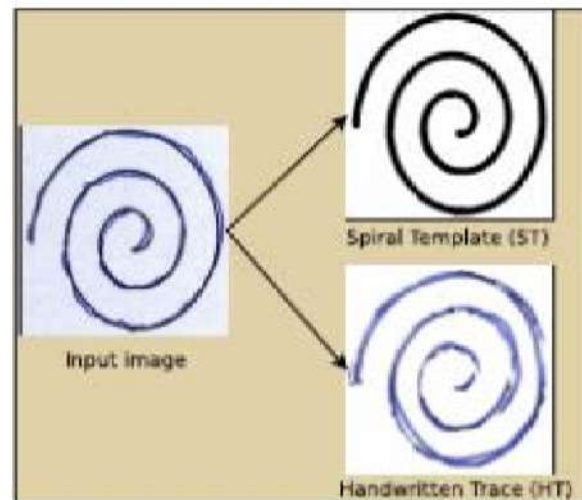


Figura 8. Imagen dividida entre el trazo real y el del paciente

Fuente: Clayton et al., (2015) *A Step Towards the Automated Diagnosis of Parkinson's Disease: Analyzing Handwriting Movements*. (p. 3)

Tercero, con los vectores característicos de las imágenes obtenidos de la etapa de extracción de características, donde se comparó el trazo real con el trazo realizado por el paciente, se aplicaron técnicas de *Machine Learning* como *Naïve Bayes*, *Support Vector Machine* con RBF Kernel.

Finalmente, se obtuvo como resultados al algoritmo *Naive Bayes* como mejor técnica de *Machine Learning* para la predicción. Con la que se obtuvo 78.9% de rendimiento y por lo cual, los autores concluyen que el utilizar *Machine Learning* sirvió para poder crear un método que sea menos costoso y ayude a las personas a poder detectar cuando una persona podría presentar la Enfermedad de Parkinson. Asimismo, concluyen que utilizando nuevas técnicas se podrían obtener mejores resultados.

En el artículo publicado en 2019, titulado *Parkinson's Disease Detection from Drawing Movements Using Convolutional Neural Networks*, cuyos autores fueron Gil, M., Montero, J. & San-Segundo, R. Inicialmente los autores hacen referencia al concepto de biometría, el cual es muy importante para la ayuda de detección temprana de enfermedades que pueden afectar a

corto o largo plazo. Además, hacen mención de que un lápiz biométrico puede ser de mucha ayuda para poder diagnosticar si una persona presenta la Enfermedad de Parkinson, ya que este podría ser un método no muy costoso. Cabe resaltar, que actualmente existe una tasa muy elevada de casos en los cuales la EP es mal diagnosticada. Por lo cual, este lápiz biométrico podría ayudar a captar los movimientos de temblor que produce esta enfermedad principalmente en las manos. Asimismo, señalan que una detección temprana puede ser de mucha ayuda para los pacientes, puesto que permitiría que los pacientes obtengan tratamientos.

La metodología consistió en 2 etapas. Primero, el conjunto de datos utilizados fueron datos públicos de imágenes de trazos en forma de espirales y de una tabla digitalizada. Cada grabación en la tabla digitalizada incluye información sobre las tres coordenadas en un plano de una imagen, presión y ángulo obtenidos con ayuda del dispositivo. Para poder realizar el preprocesamiento de las señales, se verificaron las velocidades de muestreo. Se dividió la data en dos grupos según el rango de demora de los videos, los de 0 segundos eran clasificados en una clase y los de 3 segundos en otra clase. En la Figura 9 se puede observar cómo se realizó el preprocesamiento.

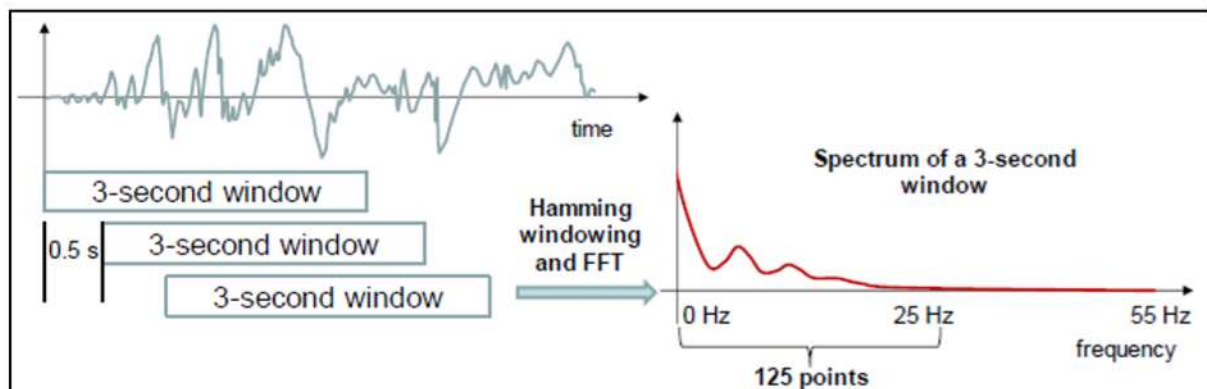


Figura 9. Procedimiento del Preprocesamiento

Fuente: Gil, M., Montero, J. & San-Segundo, R. (2019). *Parkinson's Disease Detection from Drawing Movements Using Convolutional Neural Networks*. (p. 4)

Segundo, se hizo uso de redes neuronales convolucionales, donde se consideraron 16 filtros con dimensiones 1x5, el porcentaje de validación fue del 20%. En la Figura 10 se puede observar la estructura de la red. Además, se utilizó Validación Cruzada para poder tener un mejor rendimiento, dado que la data se dividió en 5 grupos, 4 para el entrenamiento y 1 para la prueba.

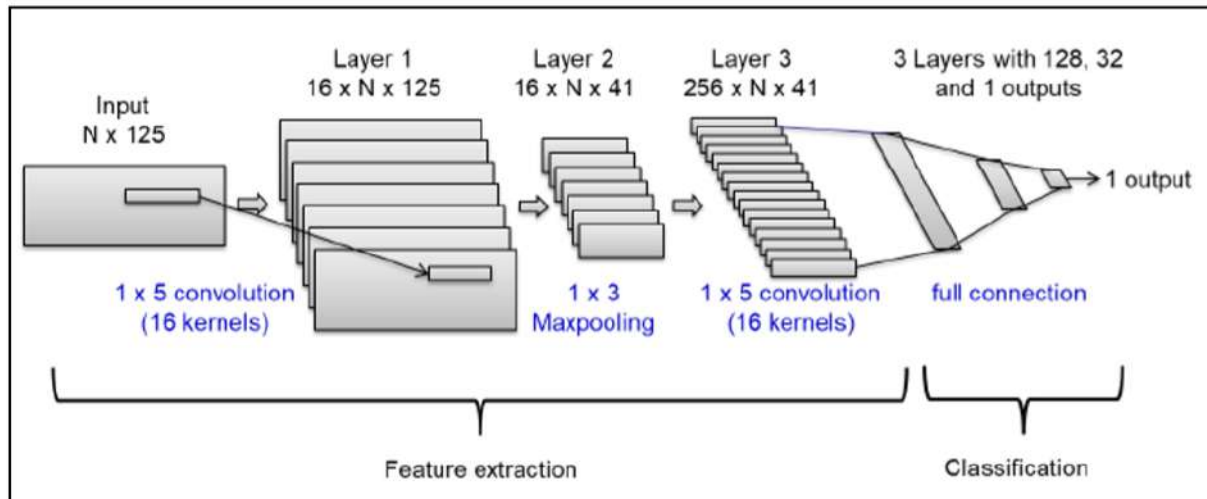


Figura 10. Estructura de la Red Convolucional

Fuente: Gil et al., (2019) *Parkinson's Disease Detection from Drawing Movements Using Convolutional Neural Networks*. (p. 5)

Como resultados, se obtuvo una precisión de 96.5%, una puntuación *F1* de 97.7% y un *AUC* de 99.2%. Con lo cual, concluyeron que el análisis de imágenes es efectivo para detectar si una persona tiende a presentar la Enfermedad de Parkinson, asimismo los experimentos realizados con *Deep Learning* fueron de mucha ayuda para incrementar la precisión de los modelos.

En el artículo publicado en 2017, titulado *A Brief Overview on Handwriting Analysis for Neurodegenerative Disease Diagnosys*, cuyos autores son De Stefano, C., Fontanellal, F., Impedovo, D., Pirlo, D. & Scotto, A. Inicialmente se habla de las enfermedades neurodegenerativas que afectan al sistema nervioso como lo es el Parkinson. Cuyo origen se basa en que todas las neuronas que están en nuestro cerebro producen energía con el fin de realizar nuestras actividades motoras. Asimismo, hacen mención de que actualmente existen diversas formas para poder diagnosticar la Enfermedad de Parkinson como el proceso de escritura, puesto que se ha identificado disgrafía en pacientes que presentaban la enfermedad de modo leve a moderada.

La metodología se dividió en 3 etapas. Primero se da una explicación del estado de arte entre los cuales analizan la escritura cinemática para cuantificar las diferencias entre las funciones motoras de la mano de pacientes que presentan la Enfermedad de Parkinson. Además, se identifica que la escritura a mano anormal es una manifestación que se presenta tempranamente, ya que puede aparecer en las primeras etapas de la enfermedad y por lo tanto

analizarlo sería un buen método para poder dar un diagnóstico. Con ayuda de la micrografía, se puede identificar diferencias en la escritura entre personas que presentan y aquellas que no presentan la EP. Además, se concluye que medidas como bradicinesia, temblor y micrografía sirven para distinguir pacientes con Parkinson de los sanos.

Segundo, en esta investigación se analizaron bases de datos con muestras que contienen datos cinemáticos y de presión. Asimismo, también contiene imágenes de espirales y de escrituras de oraciones.

Tercero, para poder distinguir a los pacientes sanos de los no sanos se utilizó tres clasificadores los cuales fueron: KNN, AdaBoost y SVM. La técnica que les ayudó a clasificar con un mejor rendimiento fue SVM.

Finalmente, concluyeron en este artículo de investigación que la escritura es una buena forma para poder detectar con anticipación si una persona presenta o no la enfermedad de Parkinson y diferentes artículos de investigación lo validan. Sin embargo, se deben tomar en cuenta todos los diferentes factores que se asocian a que una persona escriba o dibuje algo.

2.2 Bases Teóricas

2.2.1 enfermedad de parkinson

2.2.1.1 introducción

La enfermedad del Parkinson (EP) es un antiguo trastorno degenerativo, en la india se le conocía como Kampavata y Galeno, asimismo hay referencias de estos elementos en textos egipcios, etc. Antes de mostrar un análisis bibliográfico sobre las características de la escritura de una persona con la enfermedad del Parkinson (EP), es necesario detenerse para realizar una descripción sobre lo previamente mencionado, se analizarán postulados sobre las primeras bases teóricas, además de la evolución de este padecimiento en la actualidad y su influencia en la expresión escrita.

2.2.1.2 época pre-parkinson

Cierta cantidad de autores afirmaban que la enfermedad no existía antes del siglo XIX, pero algunos antecedentes como los textos vedas acreditaban la existencia de personas “temblorosas” que tenían además inconvenientes en la concentración, como tratamiento se aconsejaba en ese entonces el consumo de plantas (*Mucura Pruriens*); por otro lado un papiro egipcio menciona que un rey de edad avanzada de la dinastía XIX “...había alojado su boca, escupía continuamente”, eso hace referencia a la sialorrea parkinsoniana; asimismo un texto

sobre Aryuveda, el Charaka Samhita describe diferente variedad de temblores y algunos parálisis; en Grecia los manuscritos de Hipócrates se describe el temblor de las manos de algunas personas que se encontraban en posición de reposo. (Tagle, 2005, pág. 1-3)

2.2.1.3 primeras referencias sobre la enfermedad del parkinson

La enfermedad del Parkinson es un antiguo trastorno degenerativo, entre el siglo XVII y XIX diversos autores reconocidos describían con mucha exactitud los síntomas de la enfermedad, pero fue el Doctor James Parkinson quien recopiló sintetizó y definió mejor los síntomas de la Enfermedad de Parkinson, sus postulados fueron respaldados en observaciones clínicas realizadas a seis personas.

El proceso de entendimiento de la enfermedad es largo, primero es necesario contextualizar al Doctor Parkinson, perteneciente al periodo de la historia conocida como la edad de oro, él desde 1797 se reunió con personas relevantes y publicó diversos artículos, pero fue en 1817 cuando se realizó el documento *An Essay on Shaking Palsy* donde describía esta enfermedad como un conjunto de “Movimientos temblorosos involuntarios, con disminución de la potencia muscular en la movilidad pasiva y activa, con propensión a encorvar el tronco hacia adelante y a pesar de caminar a correr los sentidos y el intelecto no sufren mayor daño.” (García, López, Meza, 2010, pág. 359), esa monografía fue la que puso en la cima al doctor, en ella describe a seis pacientes y explica al fenómeno como una alteración en el correcto desempeño de la médula espinal que como máximo se podría extender al bulbo, en el estudio no manifestó la dureza que trae consigo este mal, eso hizo que se generen rumores sobre la falta de precisión en la apreciación de los pacientes. Fue Charcot quien analiza por primera vez la rigidez de esta parálisis agitante (nombre que le dio James Parkinson), pero la mencionada es quien acuñó el nombre enfermedad de Parkinson, con el fin de darle crédito al británico. (García, López, Meza, 2010, pág. 357-360)

2.2.1.4 la enfermedad de parkinson hoy

Al respecto, Obeso (1999), menciona que “...es una patología neurológica, crónica, e incapacitante en la que se ven alterados la forma y la velocidad de los movimientos corporales. Esta patología es producida por la pérdida de un neurotransmisor que se denomina Dopamina; ante esta pérdida, aparecen los síntomas clínicos en el Plano Motor como el temblor, rigidez, acinesia...” y Bartolomé (2002), Ostrosky (2001) hacen referencia a el “...Plano Cognitivo aparecen alteraciones de la atención, la memoria, las habilidades visoespaciales, el lenguaje, los procesos de abstracción y generalización del pensamiento y la programación de la

conducta...”, por último Adamas y Victor (1993) mencionan que “en estadios avanzados, puede aparecer demencia” (citado en Macías, 2006, pág. 48).

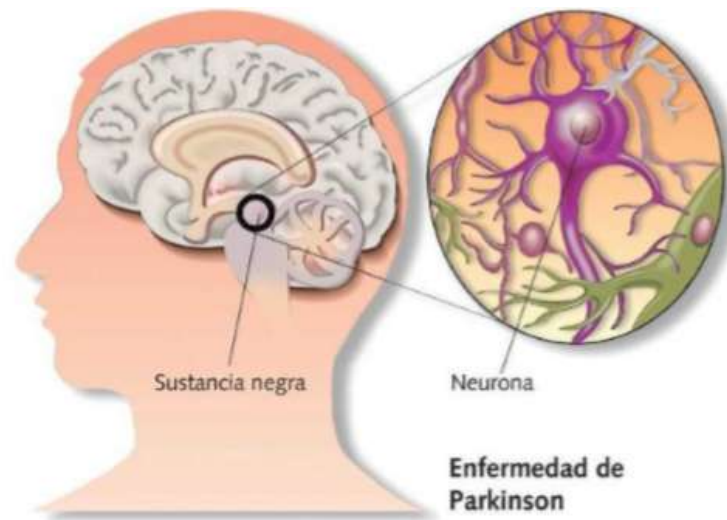


Figura 11. Sustancia negra en la EP

Fuente: Regueiro, S. & Leurona E. (2014). Ilustración de sustancia negra en la Enfermedad de Parkinson.

Esta enfermedad progresiva y crónica adquirió notoriedad a nivel mundial, se realizaron y ahondaron más estudios, haciéndolos así más minuciosos como el análisis que hace referencia a que en esta enfermedad se involucra más neurotransmisores como la serotonina, noradrenalina y acetilcolina, lo que explica a su vez otros síntomas no motores de este padecimiento.



Figura 12. Agente transmisor en la EP

Fuente: EcuRed. Agente transmisor

2.2.1.5 escritura en la enfermedad del parkinson

El temblor y rigidez de estas personas influye en cuanto a sus quehaceres diarios, lo que trae consigo una mayor dependencia hacia otra persona, ya que afecta al ámbito motor, en un estudio realizado en el año 2011 se hizo una revisión con el objetivo de analizar la bibliografía relacionada con las funciones lingüísticas y psicológicas en pacientes con la enfermedad del Parkinson (EP), en el estudio se buscaron análisis de escritura de pacientes con esta enfermedad, recolectaron datos establecidos en un rango; finalmente se registraron 74 artículos algunos tuvieron en cuenta los aspectos motores de la mecánica estructural, esta investigación hace referencia a actos y características de la escritura, se reconocen trazos, tamaño. (Rodríguez y Baquero, 2011, pág. 17-25)

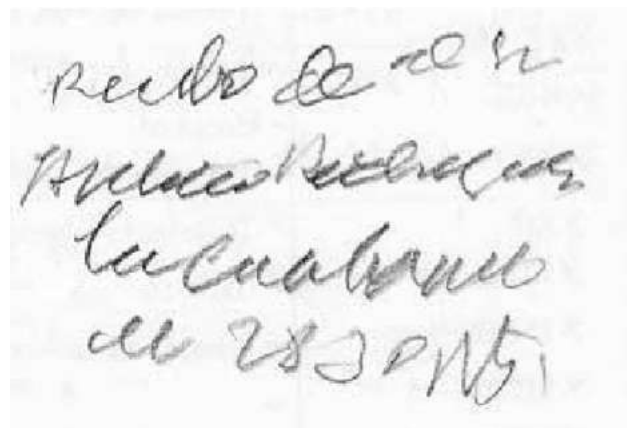


Figura 13. Ejemplo de escritura de paciente que presenta la EP

Fuente: Efisioterapia. (2012). Exploración motora.

2.2.2 visión computacional

2.2.2.1 introducción

Los humanos percibimos el mundo tridimensional que nos rodea con facilidad, ya que mediante los ojos percibimos y las imágenes se procesan en nuestro cerebro. Se puede percibir la forma, color, dimensión de las cosas y en las personas incluso las emociones. Por lo cual, a los investigadores les ha tomado mucho tiempo identificar cómo es que funciona todo este proceso y actualmente aún les resulta difícil. A pesar de dicha dificultad, actualmente el reconocimiento facial, el seguimiento de cómo camina una persona, capturar imágenes 3D, el reconocimiento de escrituras, entre otros es una realidad, pero aún no se puede comparar el cómo ve un niño de dos años, el cual capta todo su entorno, con una computadora. (Szeliski, 2010).

2.2.2.2 historia

En los 70s Minsky asigna la visión computacional como un proyecto de verano, en la cual pretende que un computador se vincule con una cámara y de ese modo describir lo que captaba. Los primeros intentos de comprender una imagen eran mediante la captura de bordes e inferir en un mundo de bloques las imágenes 3D. En los 80s las matemáticas sofisticadas fueron más tomadas en cuenta, como las pirámides de imágenes que servían para la mezcla. Además, los investigadores se enfocaron en el procesamiento de datos de rango tridimensional (adquisición, fusión, modelado y reconocimiento). En los 90s los investigadores se enfocaron en captar las imágenes en movimiento, hubo más interacción con los gráficos en la computadora y por primera vez se pudo realizar el reconocimiento facial. En el año 2000, se hacen más significativas las técnicas basadas en características lo que facilita el reconocimiento de objetos, asimismo, el desarrollo de algoritmos y el uso de técnicas de aprendizaje automático en esta época ha ido en aumento. Finalmente, debido a las grandes cantidades de datos disponibles en internet el reconocimiento de categorías de objetos es más sencillo. (Szeliski, 2010)

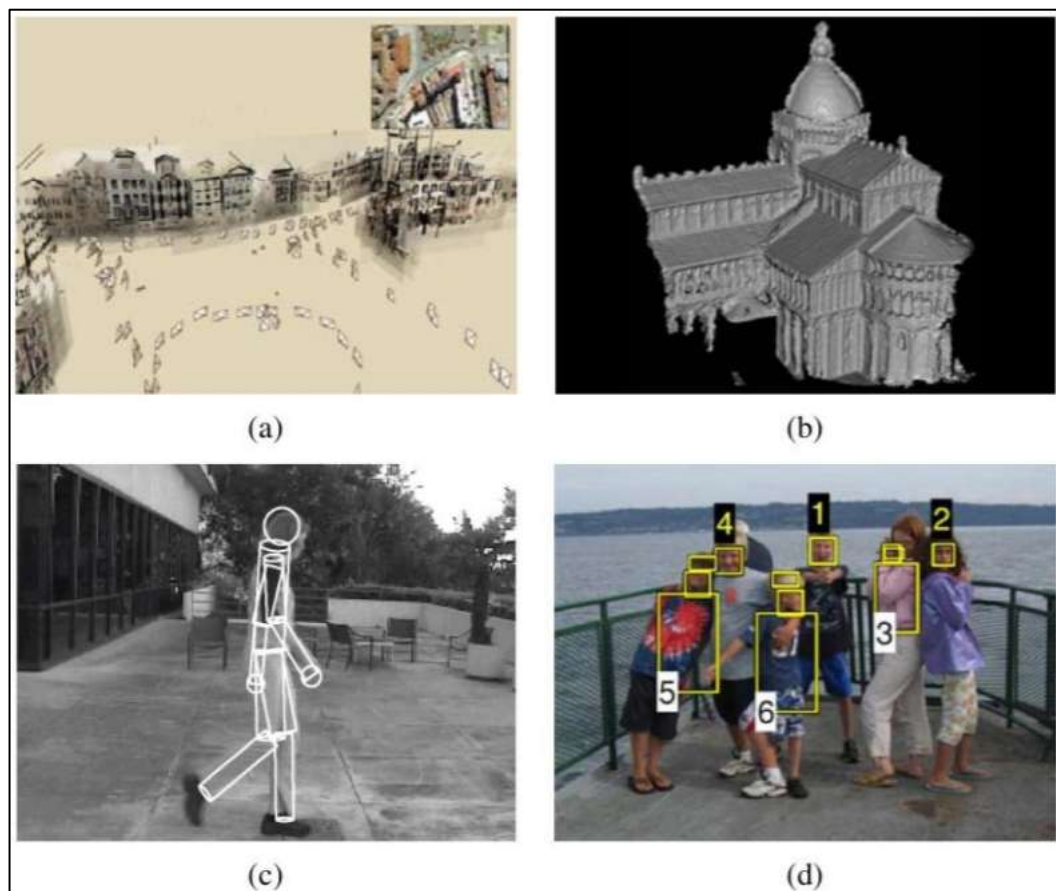


Figura 14. Ejemplos de Aplicaciones de Visión Computacional

Fuente: Szeliski, R. (2009). *Computer Vision: Algorithms and Applications*. (p.268)

2.2.2.3 etapas de la visión computacional

Según Szeliski R. (2010), en la Visión Computacional existen diferentes etapas, las cuales consisten en el desarrollo desde cómo se capta una imagen hasta la interpretación de esta. Las etapas de la visión computacional son: captación, preprocesamiento, segmentación, extracción de características, reconocimiento y clasificación y finalmente interpretación.

2.2.2.3.1 preprocesamiento

Proceso por el cual se reduce el ruido de una imagen y se realza los detalles más pequeños de una imagen. Según, Szeliski, R. (2009) en esta etapa se realizan operaciones como corrección de exposición y equilibrio de color, reducción de ruido, aumento de la nitidez, enderezamiento de la imagen, entre otros. En esta etapa se hace uso de operadores de punto que indican que el valor de cada pixel de salida depende del valor de entrada del pixel de la imagen. Los operadores son: escalamiento de brillo, contraste, transformación de color, corrección de color, etc. El operador de procesamiento de una imagen general es representado por una función que toma una o más imágenes de entrada y produce una imagen de salida. Por otro lado, la transformación de color nos permite ver a las imágenes como señales altamente correlacionadas. El preprocesamiento nos permite calcular las relaciones de color y manipular su luminiscencia. La extracción de un objeto de una imagen se llama estera, mientras que el proceso de inserción en otra imagen se llama composición. Hay un operador que se encarga de atenuar la influencia de la imagen y luego agregar los valores del color. Además, se usa la ecualización de histograma para ver el conjunto de valores de luminosidad en una imagen para posteriormente calcular valores de intensidad mínima, máxima y media. (Szeliski, 2009)

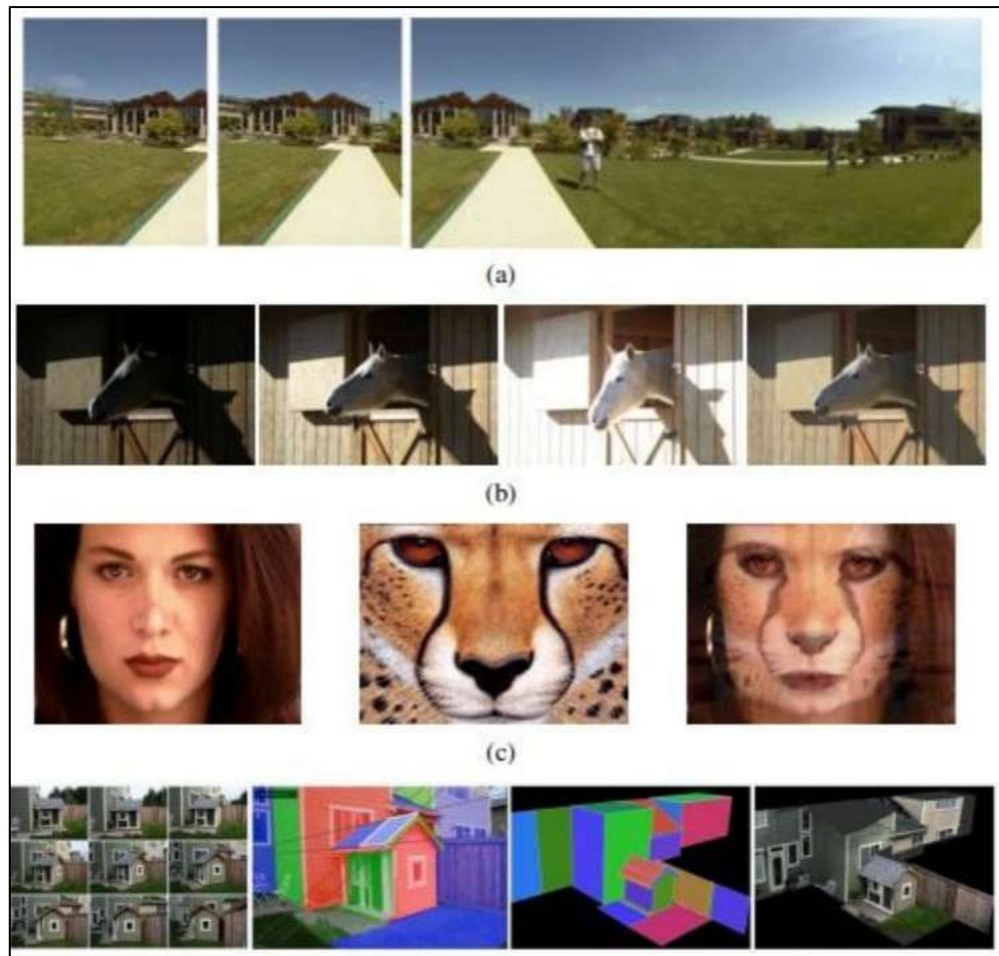


Figura 15. Características de una imagen

Fuente: Szeliski, R. (2009). *Computer Vision: Algorithms and Applications*. (p.8)

2.2.2.3.2 segmentación

En esta fase de la visión computacional se busca encontrar pixeles que van juntos. La segmentación se realiza mediante algoritmos basados en contornos activos, conjuntos de niveles, división y fusión de regiones, desplazamiento medio entre otros. Los resultados de los algoritmos de segmentación de imágenes dan resultados comparativos como métricas. En el caso de los contornos activos nos permiten localizar las curvas del límite en las imágenes. Uno de los enfoques utilizados se llama Snake es una curva bidimensional que minimiza la energía que evoluciona hacia características de la imagen como bordes fuertes. Básicamente se explica que los contornos activos permiten detectar límites gracias a sus movimientos iterativos. Otra de las técnicas de segmentación, llamado tijeras se encarga de calcular y dibujar una curva que es capaz de adherirse a los bordes de contraste de una imagen. Dicha técnica hace uso de magnitudes y orientación de gradiente para calcular costos. (Szeliski, 2009).

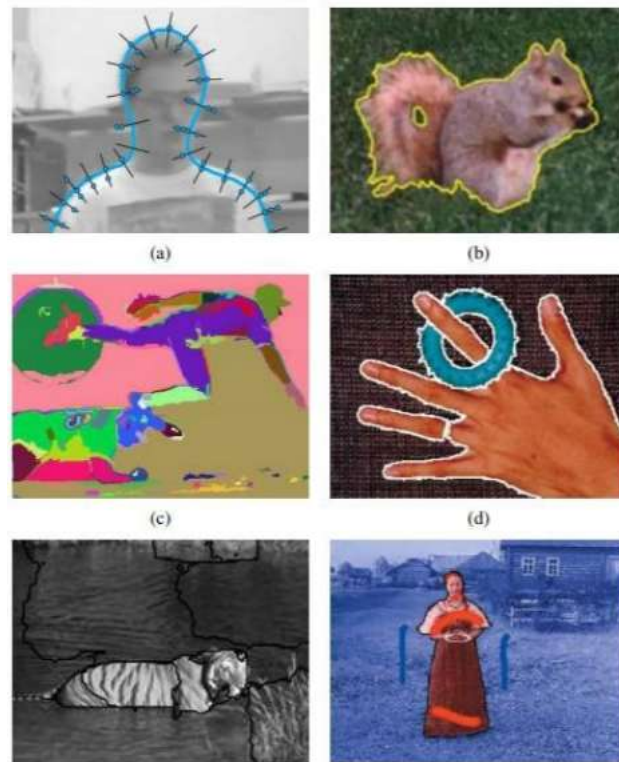


Figura 16. Técnicas de Segmentación de Imágenes

Fuente: Szeliski, R. (2009). *Computer Vision: Algorithms and Applications*. (p.268)

2.2.2.3.3 extracción de características:

En esta etapa se busca encontrar características como ubicaciones específicas en las imágenes se: picos de montaña, esquinas de las puertas o parches de nieve. Lo localizado es llamado entidades clave y son comúnmente descritos como parches de píxeles que rodean la ubicación del punto. Al final lo que se busca es agrupar dichas características para luego analizarlas para encontrar puntos de fuga. En esta etapa se puede hacer uso de funciones de puntos que permitan capturar un conjunto denso de correspondencias para alinear imágenes. Habiendo detectado las características se debe determinar su lugar de proveniencia. Posteriormente, se debe establecer coincidencias de características entre las imágenes y esto se da en dos pasos: el primero se basa en la planificación de una estrategia de emparejamiento y el segundo en la estructuración de datos para realizar la coincidencia. (Szeliski, 2009).

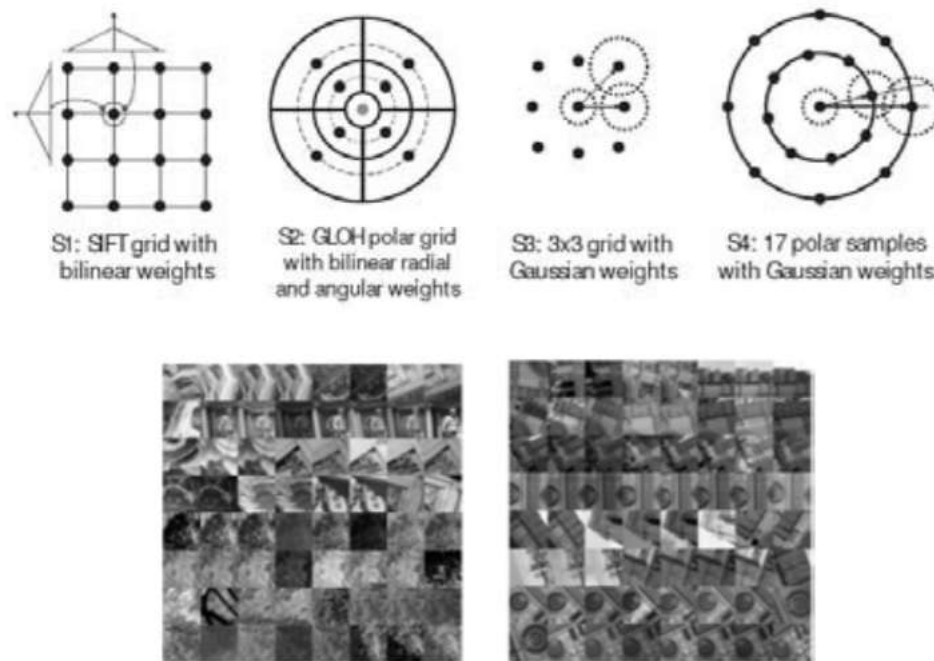


Figura 17. Extracción de características

Fuente: Szeliski, R. (2009). *Computer Vision: Algorithms and Applications*. (p.226)

2.2.2.3.4 reconocimiento y clasificación

En esta fase se busca hacer coincidir las características en diferentes imágenes mediante la geometría correspondiente. Esto se debe a que los desplazamientos de características se describen muy bien haciendo uso de geometría. Mediante el cálculo de transformaciones 2D y 3D se puede determinar la asignación de una característica a una imagen. Se busca determinar la alineación basada en características. En algunos casos la alineación que se necesitará será una 3D. La estimación pose 3D de un objeto es un ejemplo particular de alineación basada en características. (Szeliski, 2009).

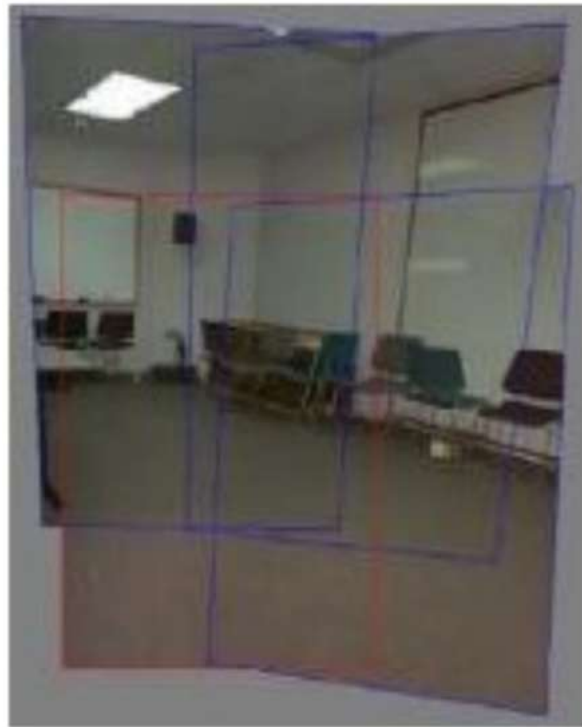


Figura 18. Clasificación de objetos

Fuente: Szeliski, R. (2009). *Computer Vision: Algoritmos and Aplicaciones*. (p.332)

2.2.2.3.5 interpretación

En esta etapa se analiza la interpretación de una escena con el sistema de visión computacional. Esta es la tarea, más compleja que puede realizar una computadora por que involucra que analice e identifique los objetos de una escena. Las computadoras son excelsas al reconstruir la forma 3D más les cuesta nombrar objetos de una escena como lo haría un ser humano. La dificultad de poder identificar reside en las variaciones extremas y lo complejo que puede ser la escena que haría que la comparación exhaustiva con una base de datos sea algo improbable. Dicho problema puede ser abordado en varias formas como lo son: detección de objetos, búsqueda de puntos característicos y el reconocimiento de categorías. Por ejemplo, la detección de objetos es muy probable que los algoritmos de búsqueda sean ineficientes y se demoren en el proceso, Una solución a este problema sería detectores con un propósito en específico. Es decir que hagamos un trabajo en conjunto de los algoritmos de búsqueda y un dispositivo digital que permite la visualización de objetos como lo es una cámara. (Szeliski, 2009).

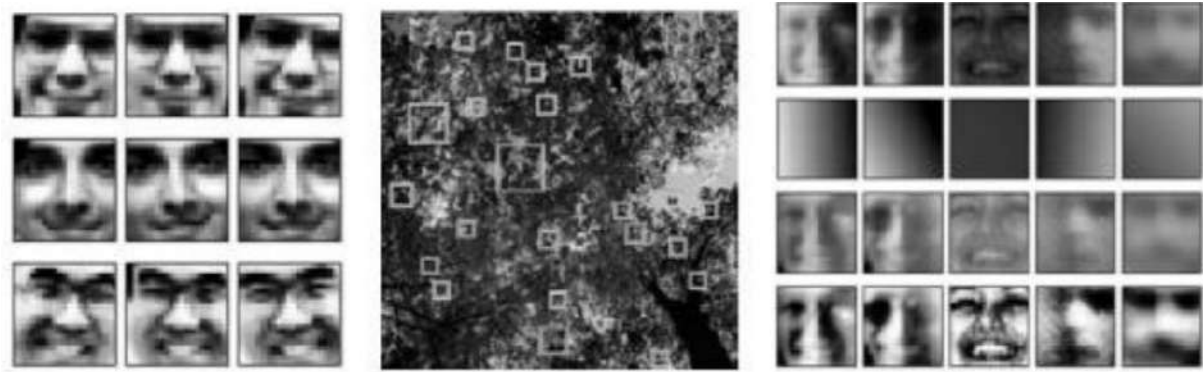


Figura 19. Detección de Rostros

Fuente: Szeliski, R. (2009). *Computer Vision: Algoritmos and Aplicaciones*. (p.332)

2.2.2.4 mejoramiento de la imagen

Sucar y Gómez (1991) sostienen que:

El objetivo de visión de bajo nivel o “procesamiento temprano” es hacer transformaciones directamente sobre la imagen para obtener información de las propiedades físicas de los objetos que están en ella y que sean de mayor utilidad para los siguientes niveles de visión. Los principales atributos que se consideran importantes para obtener de una imagen son:

- discontinuidades u orillas,
- color,
- textura,
- gradiente y profundidad. (p. 15)

2.2.2.4.1 transformaciones de intensidad

2.2.2.4.1.1 aumento lineal de contraste

Sucar y Gómez (1991) sostienen que:

Utilizando el valor de intensidad mínimo y máximo en una imagen, se puede aumentar su contraste. La idea básica es llevar el valor mínimo (min) a cero y el máximo (máx) a 255, pensando en imágenes monocromáticas (0-255). Esta transformación genera que las intensidades se espacien de acuerdo a cierto factor o pendiente; el factor para este aumento lineal de contraste es:

$$C(x, y) = \frac{I(x, y) - \min}{\max - \min} * 255 \quad (11)$$

Donde $I(x, y)$ es la imagen a procesar y $C(x, y)$ es la imagen con aumento lineal del contraste. Se puede verificar fácilmente que para $I(x, y)$ en min, $C(x, y)$ resulta cero (el numerador es cero); para $I(x, y)$ en max, $C(x, y)$ resulta en 255 (cociente 1). (p. 18)

En la Figura 20 se puede observar un ejemplo de operaciones puntuales: (a) imagen original. (b) binarización con umbral de 80. (c) negativo de la imagen original. (d) aumento lineal de contraste.

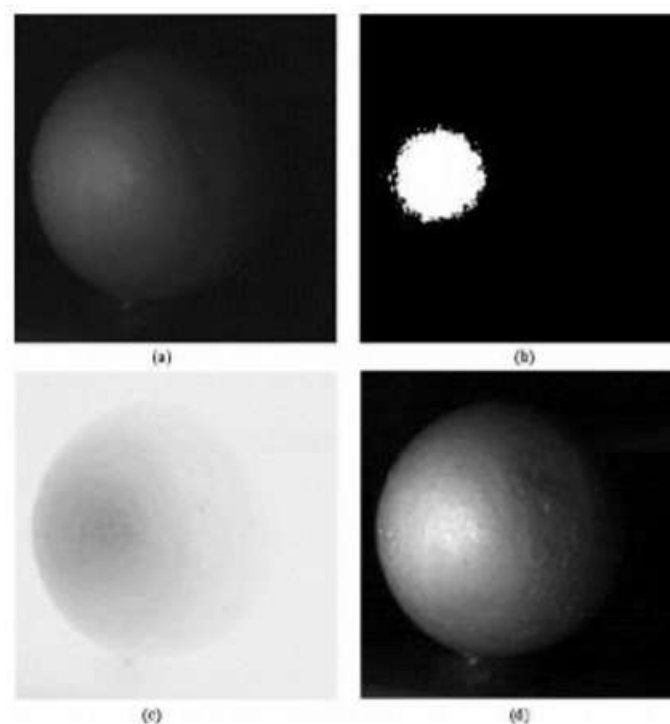


Figura 20. Ejemplo de aumento lineal de contraste

Fuente: Sucar, L. & Gómez, G. (1991). *Visión Computacional*. (p. 19)

2.2.2.4.1.2 ecualización del histograma

Sucar y Gómez (1991) sostienen que:

Un histograma de una imagen es la distribución de cada nivel de intensidad dentro de la imagen, es decir nos da un estimado de la probabilidad de ocurrencia de cada nivel de gris (r).

$$p(rk) = nk/n \quad (1)$$

Donde $p(r_k)$ es la probabilidad del nivel k , n_k es el número de pixels que toma este valor y n es el número total de pixels en la imagen. En la Figura 21 se muestra en forma gráfica el histograma de dos imágenes, una con amplio rango de intensidades y otra con un rango reducido.

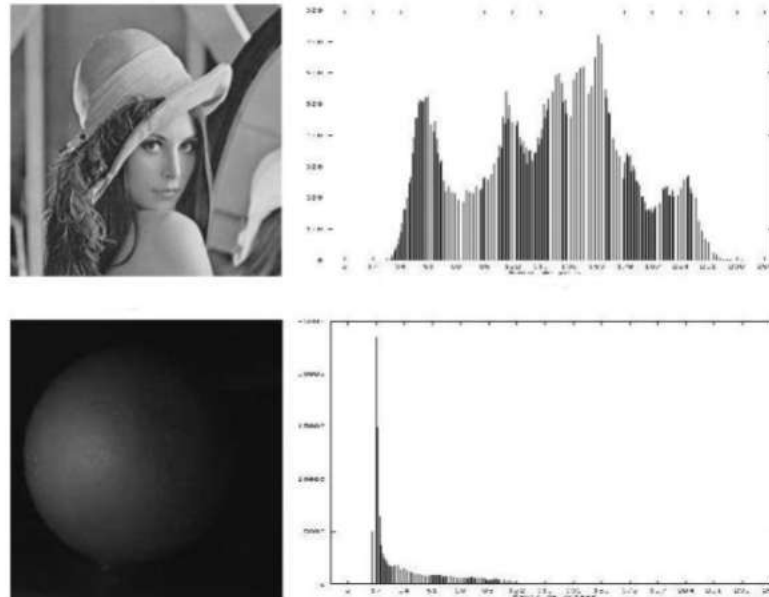


Figura 21. Histograma de dos imágenes

Fuente: Sucar, L. & Gómez, G. (1991). *Visión Computacional*. (p. 20)

2.2.2.4.2 filtros

2.2.2.4.2.1 filtro de la media aritmética

Este filtro obtiene el promedio de los pixels vecinos, es decir, todos los valores de la máscara son 1. En la Figura 22 se puede ver un ejemplo de una imagen original vs la imagen después de aplicar el filtro de la media aritmética. (Gomez y Sucar, 1991).



Figura 22. Imagen real vs Imagen con Filtro de la media aritmética

Fuente: Gomez, G. y Sucar, E. (1991). *Visión Computacional*. (p. 25)

2.2.2.4.2.2 filtro de la mediana

Substituye todos los valores de cada pixel central por la mediana de esos valores de la vecindad. (Gomez y Sucar, 1991).

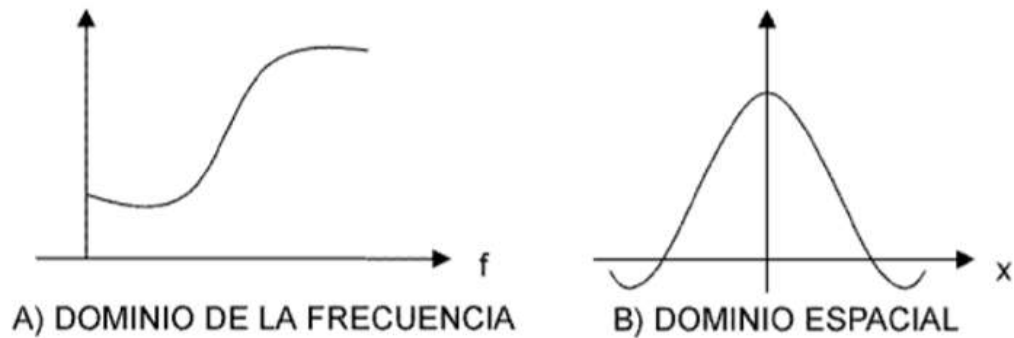


Figura 23. Aplicación del filtro de la mediana de una imagen

Fuente: Gomez, G. y Sucar, E. (1991). *Visión Computacional*. (p. 25)

2.2.2.4.2.3 filtro gaussiano

Aproximación de la distribución gaussiana a las dimensiones de una imagen. En las Figuras 24 y Figura 25 se puede observar la aplicación del filtro gaussiano. (Gomez y Sucar, 1991).

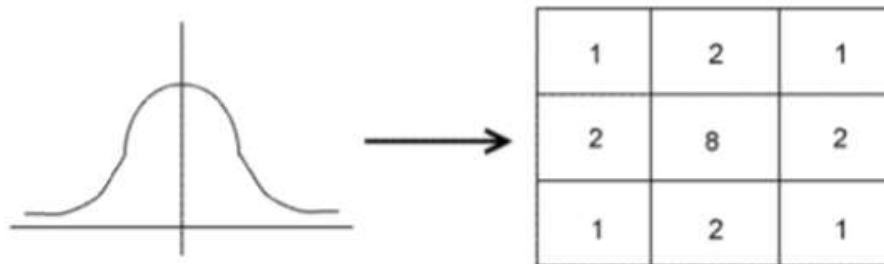


Figura 24. Máscara para filtro Gaussiano

Fuente: Gomez, G. y Sucar, E. (1991). *Visión Computacional*. (p. 24)

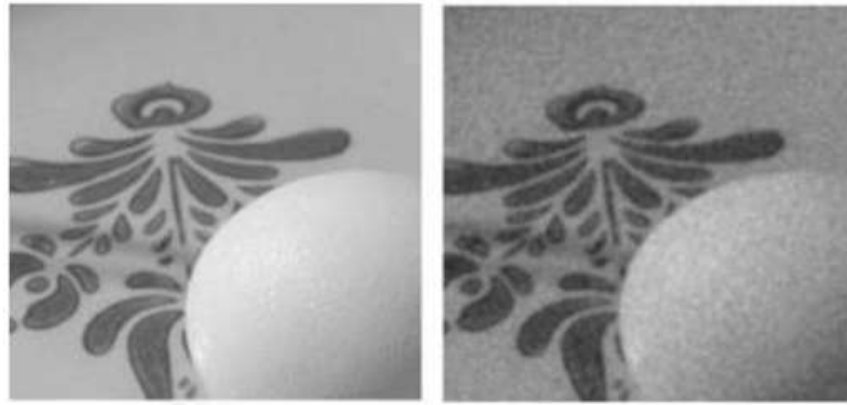


Figura 25. Imagen real vs Imagen con Filtro Gaussiano

Adaptado de: Gomez, G. y Sucar, E. (1991). *Visión Computacional*. (p. 25)

2.2.2.5 técnicas de extracción de características

2.2.2.5.1 sift

Para poder describir que es lo que hace el extractor de características SIFT se va a explicar en diferentes etapas:

- Detección extrema de espacio de escala:

En este caso, lo primero es detectar puntos clave usando el enfoque de filtrado en cascada para poder identificar la ubicaciones y escalas que puedan ser reconocidas fácilmente frente a cualquier variación de posición de una imagen, esto se puede lograr gracias a identificar un conjunto de características estables frente a todas las escalas posibles. Por lo cual, los autores determinaron usar la función gaussiana que hace referencia al espacio de escalas, este es definido como una función $L(x,y,\sigma)$, que se produce a partir de la convolución de una escala variable gaussiana, $G(x,y,\sigma)$, con una imagen de entrada, $I(x,y)$:

$$L(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \quad (2)$$

Donde $*$ es la operación de convolución en x e y , y

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x+y)/2\sigma^2} \quad (3)$$

Por lo cual, los autores plantean utilizar un espacio de escala extremo en la función de diferencia de Gauss convolucionada con la imagen, $D(x,y,\sigma)$, que puede calcularse a partir de La diferencia de dos escalas cercanas separadas por un factor multiplicativo constante k :

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \quad (4)$$

$$D(x, y, \sigma) = L(x, y, k\sigma) - G(x, y, \sigma) \quad (5)$$

En la siguiente Figura 26 se puede observar la diferencia

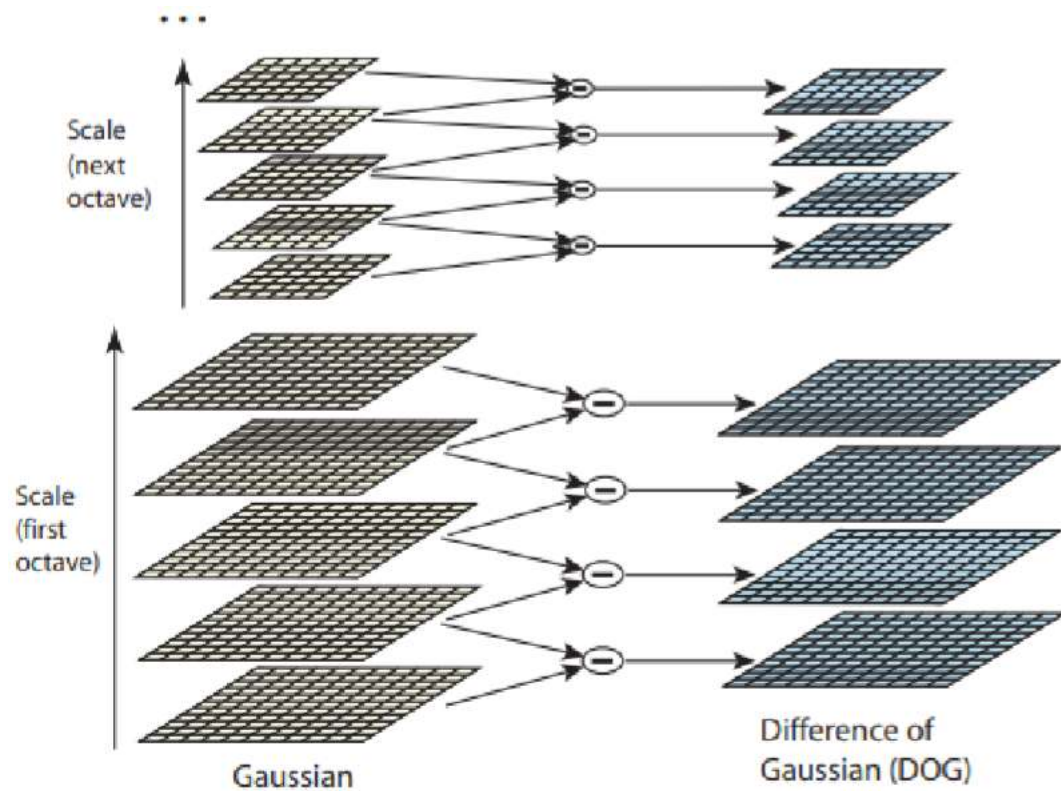


Figura 26. Gaussian vs Diferencia de Gaussian

Fuente: Lowe, D. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. (p.6)

Las imágenes gaussianas adyacentes se restan para producir las imágenes de diferencia de gaussianas a la derecha. Después de cada octava, la imagen gaussiana se muestrea hacia abajo por un factor de 2, y el proceso se repite. (Lowe, 2004)

- Localización de puntos clave:

Una vez que se ha encontrado un candidato clave comparando un píxel con sus vecinos, el siguiente paso es realizar un ajuste detallado a los datos cercanos para la ubicación, escala y relación de curvaturas principales. Esta información permite rechazar puntos que tienen poco contraste (y, por lo tanto, son sensibles al ruido) o están mal localizados a lo largo de un borde. (Lowe, 2004)

En la Figura 27 se puede observar la selección de puntos característicos. En (a) está la imagen original de 233x189 píxeles, en (b) las ubicaciones iniciales de 832 puntos clave en máximos y mínimos de la función de diferencia de Gauss. Los puntos clave se muestran como

vectores que indican escala, orientación y ubicación. En (c), después de aplicar un umbral de contraste mínimo, quedan 729 puntos clave. En (d), los 536 puntos finales que permanecen después de un umbral adicional en la relación de curvaturas principales.

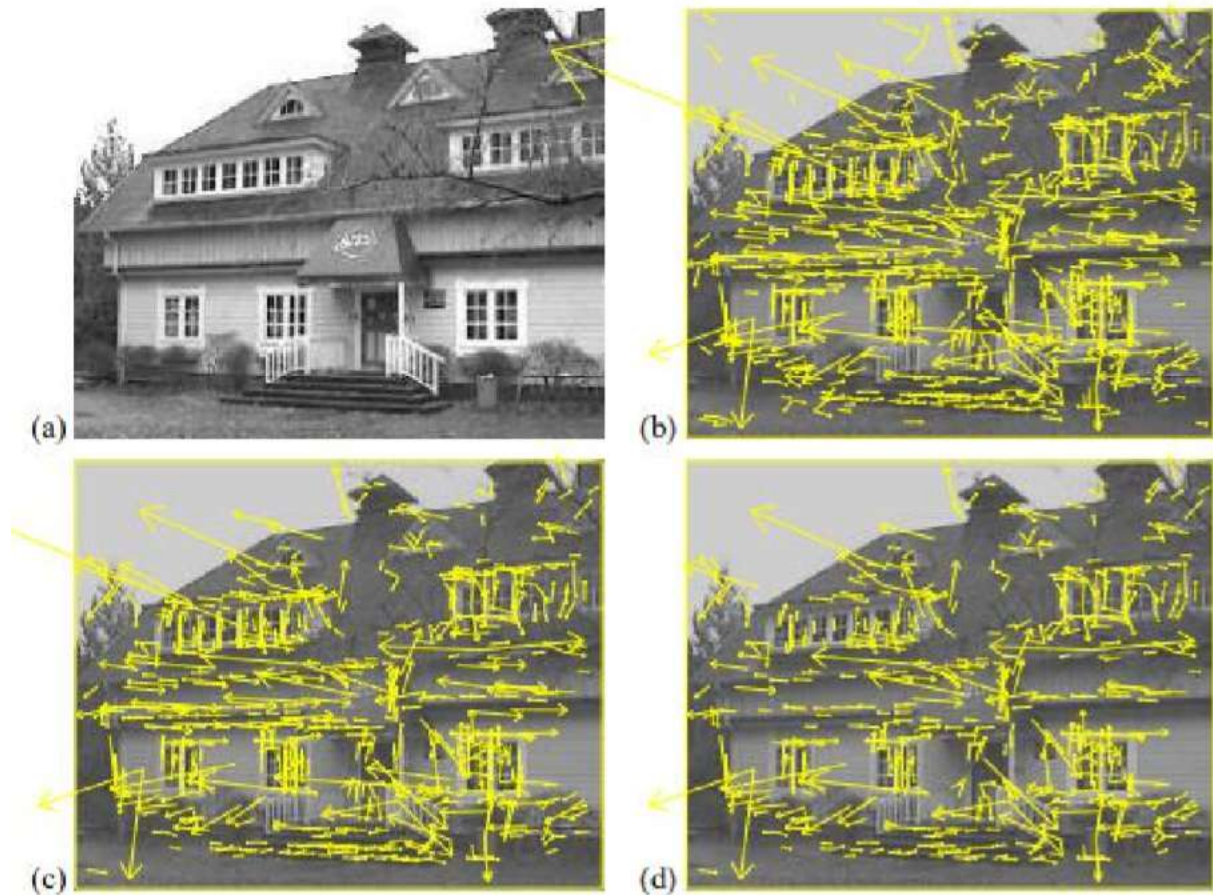


Figura 27. Selección de puntos característicos

Fuente: Lowe, D. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. (p.11)

- Asignación de orientación:

Al asignar una orientación coherente a cada punto clave en función de las propiedades de la imagen local, el descriptor del punto clave se puede representar en relación con esta orientación y, por lo tanto, lograr la invariabilidad en la rotación de la imagen. (Lowe, 2004)

La escala del punto clave se utiliza para seleccionar la imagen suavizada de Gauss, L , con la escala más cercana, de modo que todos los cálculos se realicen de forma invariable. Para cada muestra de imagen, $L(x, y)$, a esta escala, la magnitud del gradiente, $m(x, y)$, y la orientación, $\theta(x, y)$, se calcula previamente utilizando diferencias de píxeles:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (6)$$

$$\theta(x, y) = \tan^{-1} \sqrt{(L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y))} \quad (7)$$

- Descriptor de punto clave:

Las operaciones anteriores han asignado una ubicación, escala y orientación de imagen a cada punto clave. Estos parámetros imponen un sistema de coordenadas 2D local repetible en el que se describe la región de la imagen local y, por lo tanto, proporcionan invariancia a estos parámetros. El siguiente paso es calcular un descriptor para la región de la imagen local que sea altamente distintivo pero que sea lo más invariable posible a las variaciones restantes, como el cambio de iluminación o el punto de vista 3D. (Lowe, 2004)

En la Figura 28 se muestra que se crea un descriptor de punto clave calculando primero la magnitud y la orientación del gradiente en cada punto de muestra de imagen en una región alrededor de la ubicación del punto clave, como se muestra a la izquierda. Estos son ponderados por una ventana gaussiana, indicada por el círculo superpuesto. Estas muestras se acumulan en histogramas de orientación que resumen el contenido en subregiones 4x4, como se muestra a la derecha, con la longitud de cada flecha correspondiente a la suma de las magnitudes de gradiente cerca de esa dirección dentro de la región. Esta figura muestra una matriz de descriptores de 2x2 calculada a partir de un conjunto de muestras de 8x8, mientras que los experimentos en este documento usan descriptores de 4x4 calculados a partir de una matriz de muestras de 16x16. (Lowe, 2004)

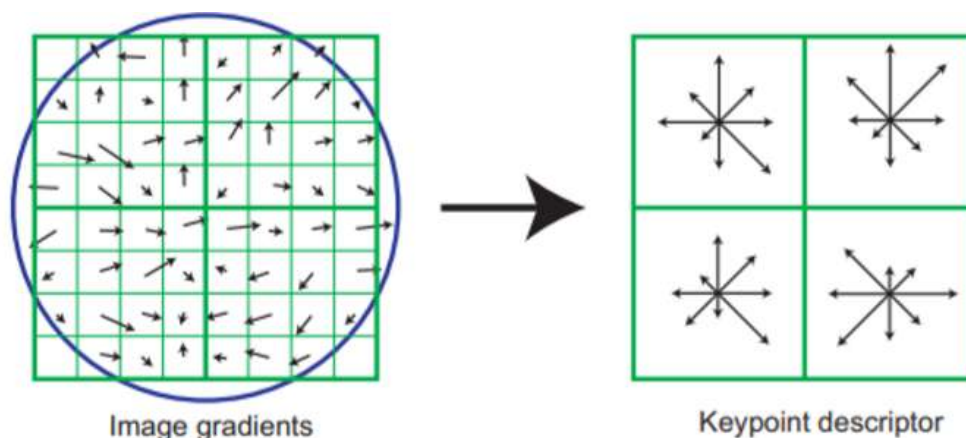


Figura 28. Descriptor de un punto clave

Fuente: Lowe, D. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. (p.15)

2.2.2.5.2 surf

SURF (Speeded Up Robust Features), los autores se basan en el detector en base a la matriz de Hessiana debido a su buen rendimiento en tiempo de cálculo y precisión. Sin

embargo, en lugar de utilizar una medida diferente para seleccionar la ubicación y la escala, confiaron en el determinante del Hessian para ambos. Dado un punto $x = (x, y)$ en una imagen I , la matriz de Hesse $H(x, \sigma)$ en x a escala σ se define de la siguiente manera:

$$H(x, \sigma) \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (8)$$

donde $L_{xx}(x, \sigma)$ es la convolución de la derivada de segundo orden gaussiana $\partial^2/\partial x^2 g(\sigma)$ con la imagen I en el punto x , y de manera similar para $L_{xy}(x, \sigma)$ y $L_{yy}(x, \sigma)$. (Bay, Tuytelaars y Van Gool, 2006)

SURF consiste en 2 etapas:

- Asignación de Orientación:

Para ser invariante a la rotación, identificamos una orientación reproducible para los puntos de interés. Para ese propósito, primero calculamos las respuestas de la Haar-wavelet en la dirección x e y , que se muestra en la Figura 29, y esto en una vecindad circular de radio $6s$ alrededor del punto de interés, con s la escala en la que se detectó el punto de interés. (Bay, Tuytelaars y Van Gool, 2006)

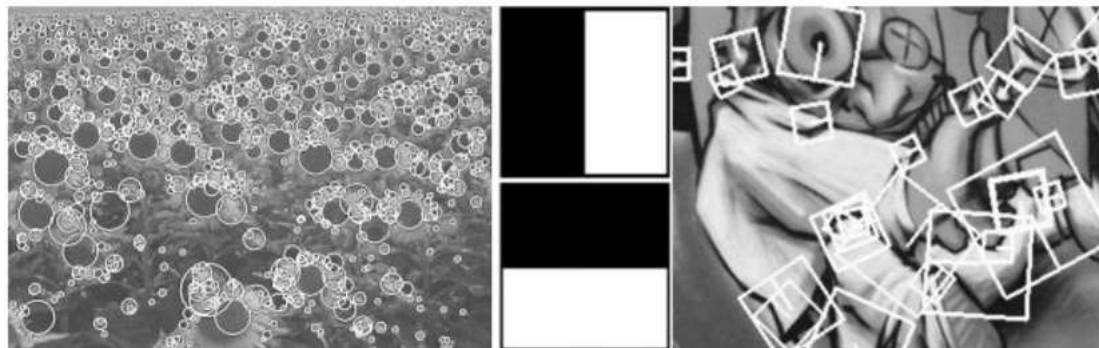


Figura 29. Respuestas de la Haar-wavelet

Fuente: Bay et al., (2006). *SURF: Speeded Up Robust Features*. (p. 6)

- Componentes descriptores:

Para la extracción del descriptor, el primer paso consiste en construir una región cuadrada centrada alrededor del punto de interés y orientada a lo largo de la orientación seleccionada en la sección anterior. Para aumentar la robustez frente a las deformaciones geométricas y los errores de localización, las respuestas dx y dy se ponderan primero con un gaussiano ($\sigma = 3.3s$) centrado en el punto de interés. Luego, las respuestas wavelet dx y dy se

resumen sobre cada subregión y forman un primer conjunto de entradas para el vector de características. Para traer información sobre la polaridad de los cambios de intensidad, también extraemos la suma de los valores absolutos de las respuestas, $|dx|$ y $|dy|$. Por lo tanto, cada subregión tiene un vector descriptor de cuatro dimensiones v para su estructura de intensidad subyacente $v = (dx, dy, |dx|, |dy|)$. Esto da como resultado un vector descriptor para todas las subregiones 4×4 de longitud 64. (Bay, Tuytelaars y Van Gool, 2006)

La Figura 30 muestra las propiedades del descriptor para tres patrones de intensidad de imagen claramente diferentes dentro de una subregión. Se puede decir que las entradas del descriptor de una subregión representan la naturaleza del patrón de intensidad subyacente. Izquierda: en el caso de una región homogénea, todos los valores son relativamente bajos. Medio: en presencia de frecuencias en la dirección x , el valor de $|dx|$ es alto, pero todos los demás siguen siendo bajos. Si la intensidad aumenta gradualmente en la dirección x , ambos valores dx y $|dx|$ son altos. (Bay, Tuytelaars y Van Gool, 2006)

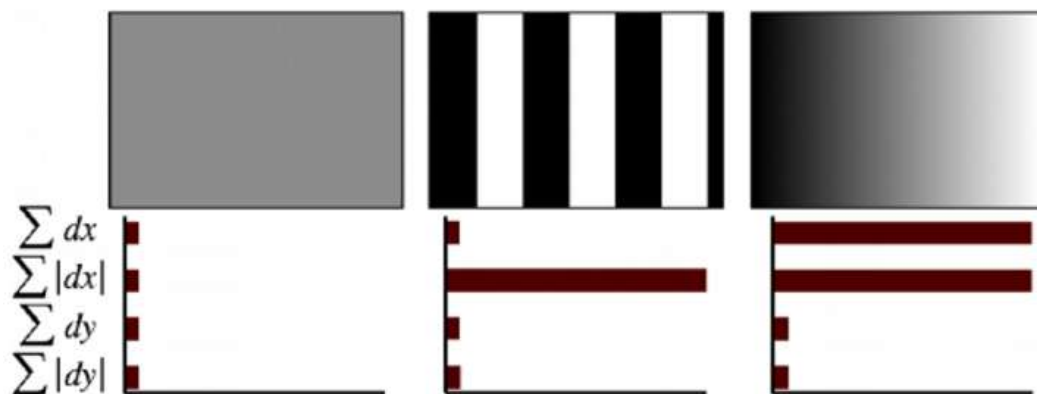


Figura 30. Propiedades del descriptor para tres patrones de intensidad de imagen

Fuente: Bay et al., (2006). *SURF: Speeded Up Robust Features*. (p. 8)

2.2.2.2.3 orb

ORB es la combinación entre el detector de puntos claves FAST y el descriptor BRIEF. El funcionamiento de ORB se basa en usar FAST para poder encontrar los puntos clave en base a las esquinas de los elementos de una imagen aplicando la Medida de las Esquinas de Harris. Sin embargo, un problema de FAST es que no calcula la orientación. Por lo que a los autores se les ocurrió calcularla mediante la intensidad ponderada del centroide del parche con la esquina ubicada en el centro. Es decir, la orientación se da desde el punto de la esquina hacia el centroide. Asimismo, el centroide de intensidad supone que la intensidad de una esquina está

desplazada de su centro, y este vector puede usarse para imputar una orientación. Los momentos de un parche son definidos como:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (9)$$

y con estos momentos se puede encontrar el centroide:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (10)$$

Se puede construir un vector desde el centro de las esquinas, O, hasta el centroide, OF ~. La orientación del parche simplemente es:

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (11)$$

donde atan2 es la versión de arctan que reconoce el cuadrante. En la Figura 31 se puede observar cómo es que el resultado de coincidencia típico usando ORB en imágenes del mundo real con cambio de punto de vista. Las líneas verdes son coincidencias válidas; los círculos rojos indican puntos sin igual. (Rublee, Rabaud, Konolige y Bradski, 2011).

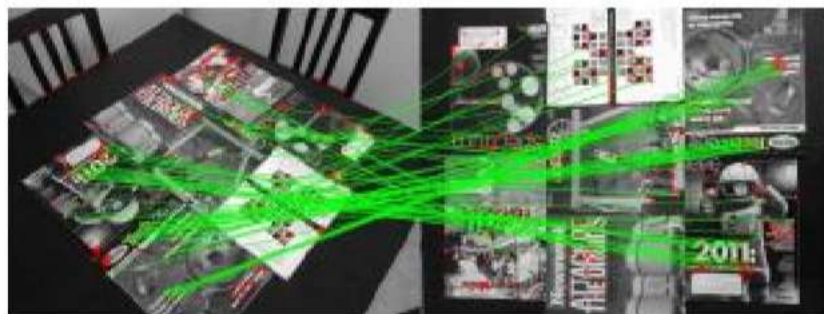


Figura 31. Resultado coincidente utilizando ORB en imágenes del mundo real con cambio de punto de vista.

Fuente: Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. (2011). *ORB: an efficient alternative to SIFT or SURF*. (p. 1)

Para los descriptores se planteó usar BRIEF, pero como ya se sabía, este presentaba un mal funcionamiento con la rotación y es así que ORB dirige a BRIEF mediante los puntos claves. Se dice que ORB es binario debido a que para cualquier conjunto de características de n pruebas binarias se toma su respectiva ubicación (x,y) para orientarla hacia el parche y poder hacer la rotación. La estrategia de ORB es realizar una búsqueda entre todas las pruebas binarias posibles para encontrar la que presenta mayor varianza y un valor cercano a 0.5. A todas las modificaciones hechas a BRIEF se le denomina rBRIEF. Finalmente, se concluye que

ORB tiene un poder de ejecución más rápido que SIFT y SURF, un mejor funcionamiento y no está patentado a diferencia de los descriptores mencionados SIFT y SURF. (Rublee et al., 2011).

2.2.2.2.4 hog

Los descriptores HOG presentan un performance equivalente al descriptor SIFT, con la excepción de que desarrollados bajo el enfoque de un grid denso que presenta celdas espaciadas de forma uniforme y que hace uso normalización del contraste de superposición local para mejorar el performance de los descriptores. (Dalal y Triggs, 2005).

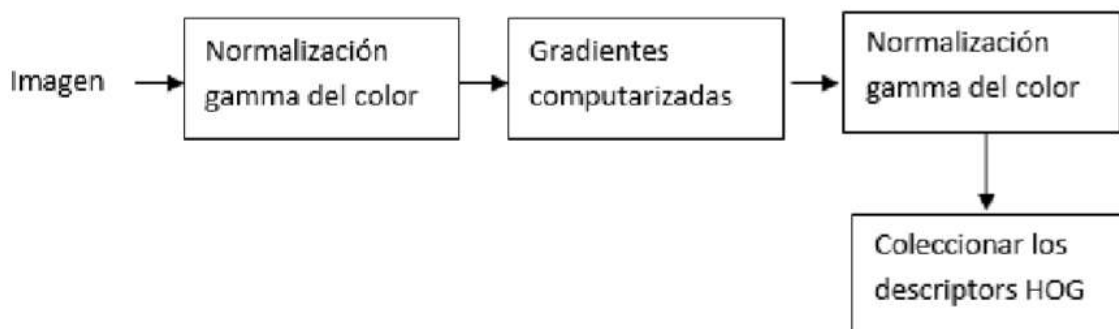


Figura 32. Metodología del HOG

Fuente: Dalal, N. & Triggs, B. (2005). *Histograms of Oriented Gradients for Human Detection*. (p. 2)

El funcionamiento del algoritmo HOG empieza con la normalización gamma y el color. Es decir, las imágenes son divididas entre regiones de mismo espacio acumulando en el proceso histogramas de direcciones de gradiente sobre los píxeles de cada región. Además, para mejorar la iluminación, es necesario hacer uso de la normalización del contraste. HOG computa la estructura de la gradiente de los contornos, siendo esta una característica de la forma de la imagen. En la Figura 33 se puede observar un ejemplo de HOG.

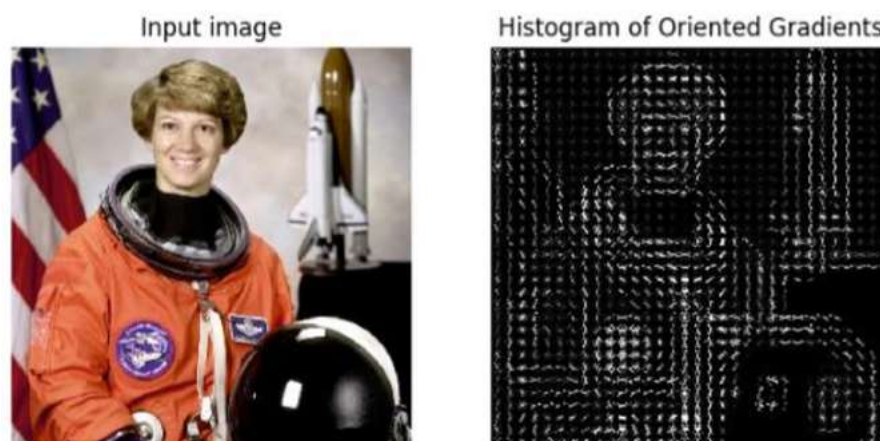


Figura 33. Ejemplo de HOG

Fuente: Scikit-image. (s.f.). *Histogram of Oriented Gradients*.

2.2.3 inteligencia artificial

2.2.3.1 introducción

El ser humano es llamado científicamente Homo Sapiens, ya que como seres inteligentes y esta inteligencia es muy importante para nosotros. A lo largo de los años, el ser humano ha tratado de entender cómo es que pensamos, entendemos, percibimos, predecimos, entre otros. La Inteligencia Artificial o IA es uno de los campos más nuevos en la ciencia, no solo intenta entender, sino que va más allá y construye entidades inteligentes. Actualmente, abarca muchos subcampos que van de lo general a lo específico como matemáticas, salud, escribir poesías, conducir un automóvil, entre otros. (Russel & Norvig, 2010).

2.2.3.2 historia

En 1956 se propuso por primera vez estudiar la inteligencia, es así como John McCarthy, Minsky, Shannon, Rochester, entre otros, quienes eran estudiantes y colegas del MIT, CMU, Stanford e IBM se juntaron en Dartmouth College donde los principales resultados fueron acuñar el nombre de la Inteligencia Artificial y definir a la IA como disciplina independiente netamente informática. En los siguientes años la IA siguió siendo estudiada en donde tenemos ejemplos como las redes neuronales. Desde los años 80's hasta la actualidad, se viene dando a la IA como industria dado que las empresas construyen sistemas expertos de visión artificial, robots, software y hardware especializado. Seguidamente en 1987, la IA adoptó el método científico para su desarrollo, es decir implementar una metodología, hipótesis, entre otros. Desde 1995 hasta la actualidad se introduce el concepto de agentes, los cuales se van a mencionar más adelante. A partir del 2001, ya se empieza a pensar sobre la disponibilidad de bases de datos gigantes, es decir cómo poder procesarlas y almacenarlas. (Russel & Norvig, 2010).

2.2.3.3 definición

Entre las definiciones más importantes que describen los autores están las siguientes, las cuales se pueden observar en la Tabla 5. (Russel & Norvig, 2010).

Tabla 5.

Definiciones de IA del punto de vista de diferentes autores

Sistemas que:	Como Humanos	Racionalmente
---------------	--------------	---------------

Piensan	“El nuevo y excitante esfuerzo de hacer que los computadores piensen... Máquinas con mente, en el más amplio sentido literal.” (Haugeland,1985)	“El estudio de los cálculos que hacen posible percibir, razonar y actuar” (Winston, 1992)
Actúan	“El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia.” (Kurzweil, 1990)	“La inteligencia Computacional es el estudio del diseño de agentes inteligentes.” (Poole et al, 1998)

Fuente: Russel, S. & Norvig, P. (2010). *Artificial Intelligence A modern Approach*. (3ra ed.). Pearson Education, Inc. (p. 2)

2.2.3.3.1 pensar como humano (el enfoque de modelo cognitivo)

El enfoque del Modelo Cognitivo es una combinación entre la IA y la Psicología, ya que para saber si una máquina piensa como un humano es necesario saber primero cómo es que pensamos los humanos. Por lo cual, los autores proponen penetrar en la mente del humano mediante la Introspección, es decir observando los pensamientos propios y mediante la Psicología, es decir observando los pensamientos de otros. (Russel & Norvig, 2010)

2.2.3.3.2 pensar racionalmente (el enfoque de las leyes del pensamiento)

En el pasado Aristóteles, fue uno de los primeros en codificar “la manera correcta de pensar”. Sus silogismos, son estructuras de argumentación que a partir de premisas correctas llevan a una conclusión correcta. Por ejemplo, “Todos los hombres son mortales. Sócrates es un hombre. Por lo cual, Sócrates es mortal” (Russel & Norvig, 2010)

2.2.3.3.3 comportarse como humano (el enfoque del test de turing)

Fue propuesta por Alan Turing en 1950. Lo cual se resume en que una computadora era denominada inteligente siempre y cuando un operador que hacía preguntas, a un humano y a una computadora, era incapaz de distinguir las respuestas entre ambos. (Russel & Norvig, 2010)

2.2.3.3.4 comportarse racionalmente (el enfoque de los agentes)

Según Russel y Norvig (2010) señalan que “Un agente es una entidad capaz de percibir su entorno por medio de sensores, y actuar en su entorno por medio de actuadores” (p. 34).

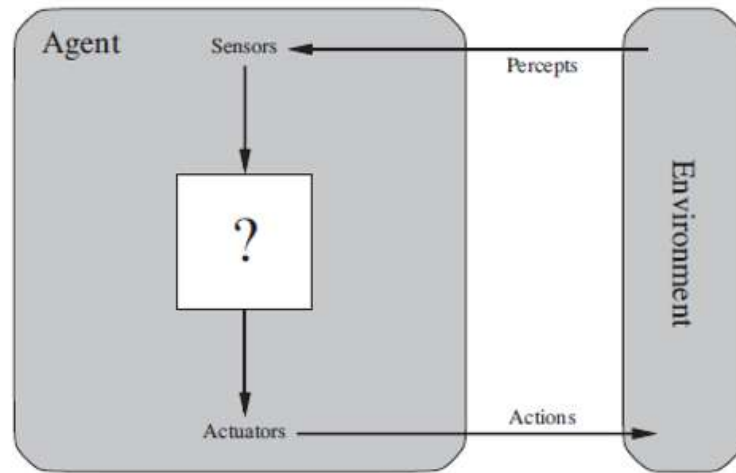


Figura 34. Interacción entre los agentes y entornos

Fuente: Russel, S. & Norvig, P. (2010). *Artificial Intelligence A modern Approach*. (3ra ed.). Pearson Education, Inc. (p. 35)

En la Figura 11 se puede observar que los agentes interactúan con los entornos a través de sensores y actuadores. La función de los agentes se basa en que para toda una secuencia de percepciones se ejecuta una acción, es decir se determina completamente el comportamiento del agente. En la Figura 35 se puede observar una aspiradora: El entorno son los cuadros A y B. Las percepciones son en qué cuadro está la aspiradora y si hay suciedad o no. Y finalmente, las acciones serían moverse a la derecha/izquierda, aspirar la suciedad o nada. Por lo cual, se concluye que un agente va a realizar una determinada acción frente a las percepciones. Además, un Agente Racional es aquel que hace lo correcto, es decir aquel que obtenga el mejor resultado. (Russel & Norvig, 2010)

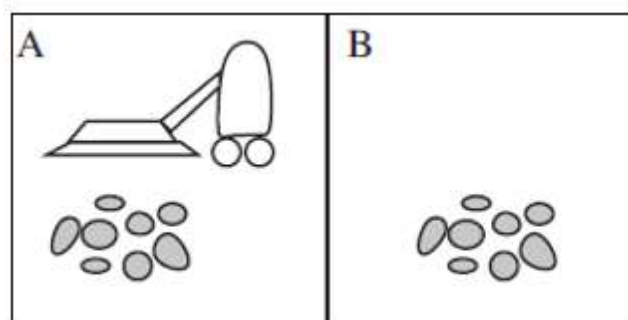


Figura 35. Acciones de una aspiradora con solo dos entornos

Fuente: Russel, S. & Norvig, P. (2010). *Artificial Intelligence A modern Approach*. (3ra ed.). Pearson Education, Inc. (p. 36)

2.2.3.4 aplicaciones

Entre las principales aplicaciones donde se puede encontrar la IA están: Robótica, Finanzas, Salud, Escritura, Economía, Software, Educación, entre otros. En la Figura 36 se puede observar números escritos a mano del 0 al 9 que pueden ser detectados con ayuda de la IA. Asimismo, en otras aplicaciones también existen la predicción del precio de las casas a futuro, creación de robots inteligentes, reconocimiento de voz, filtros automáticos de spam en correos, planeamiento de logística, entre otros. (Russel & Norvig, 2010)

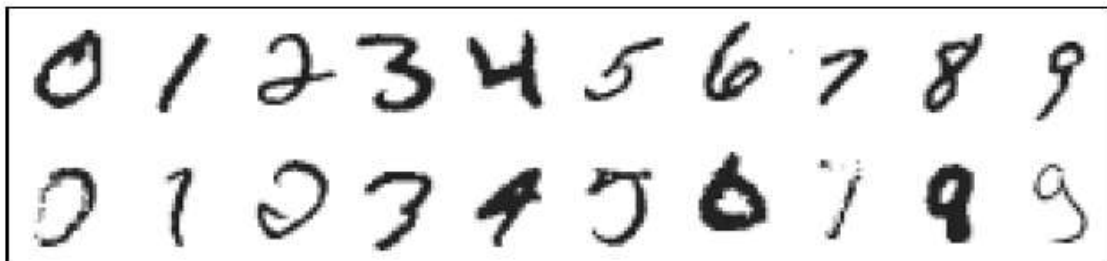


Figura 36. Números del 0 a 9 escritos a mano.

Fuente: Russel, S. & Norvig, P. (2010). *Artificial Intelligence A modern Approach*. (3ra ed.). Pearson Education, Inc. (p. 754)

2.2.3.5 machine learning

Harrington (2012) señaló que “Con Machine Learning se puede obtener información de un conjunto de datos; vamos a pedirle a la computadora que tenga sentido a partir de ellos. Esto entendemos por aprendizaje, no la memorización del ciborg, y no la creación de seres sintientes.” (p. 3) Además, actualmente *Machine Learning* es usado en todas partes, por ejemplo, si es cumpleaños de alguien y al buscar en internet te salen las 10 páginas más divertidas, luego la información que buscaste la encuentras en tu bandeja de correos Spam, te diriges a una tienda y compras pañales y la computadora genera un cupón de descuento para comprar cerveza, dado que las personas que compran pañales tienden a comprar cerveza. Además, cuando vas a un banco y la computadora te dice cuál sería tu mejor inversión o a qué tipo de crédito acceder, entre otros. (Harrington, 2012)

Machine Learning posee tres tipos de aprendizaje: Aprendizaje Supervisado, Aprendizaje no Supervisado y Aprendizaje por Reforzamiento. De los cuales, entraré en detalle con los dos primeros.

2.2.3.5.1 aprendizaje no supervisado

Harrington (2012) señaló que “En el Aprendizaje no Supervisado, no hay una etiqueta o valor objetivo para los datos. Una tarea donde se agrupamos elementos similares se conoce como Clustering... Otra tarea puede ser reducir los datos desde muchas variables hasta un número pequeño.” (p.10)

2.2.3.5.1.1 k-means

Harrington (2012) propone que:

K-Means es un algoritmo que encontrará k clusters para un conjunto de datos dado. El número de clústeres k está definido por el usuario. Cada grupo se describe por un único punto conocido como centroide. Centroide significa que está en el centro de todos los puntos del clúster. El algoritmo k-means funciona así. Primero, los k centroides se asignan aleatoriamente a un punto. A continuación, cada punto del conjunto de datos se asigna a un clúster. La tarea está hecha encontrando el centroide más cercano y asignando el punto a ese grupo. Después de este paso, todos los centroides se actualizan tomando el valor medio de todos los puntos en ese grupo. (p. 208-209).

Aplicando el algoritmo de K-Means se puede visualizar en la Figura 37 varios conjuntos de datos agrupado, provenientes de una base de datos, según características similares.

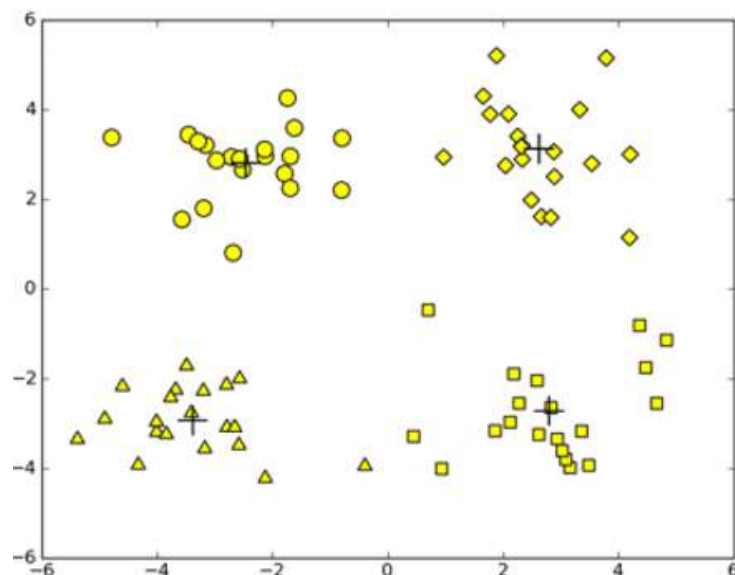


Figura 37. Clusters resultantes al aplicar K-Means

Fuente: Harrington, P. (2012). *Machine Learning in Action*. Manning Publications Co. (p. 212)

2.2.3.5.2 aprendizaje supervisado

Harrington (2012) señaló que “La clasificación y la regresión son ejemplos de aprendizaje supervisado. Este conjunto de problemas se conoce como supervisado porque estamos diciéndole al algoritmo qué predecir.” (p. 10) Cuando se quiere clasificar, el objetivo es predecir a que clase pertenece una instancia de datos. Sin embargo, la regresión tiene como bojetivo predecir un valor numérico. (Harrington, 2012)

2.2.3.5.2.1 knn

Harrington (2012) sostiene que:

Tenemos un conjunto existente de datos, de ejemplo, nuestro conjunto de entrenamiento. Tenemos etiquetas para todos estos datos: sabemos en qué clase debe caer cada parte de los datos. Cuando se nos da una nueva pieza de datos sin una etiqueta, comparamos esa nueva pieza de datos a los datos existentes, cada pieza de datos existentes. Luego tomamos los datos más similares (los vecinos más cercanos) y miramos sus etiquetas. Observamos los primeros k datos más similares de nuestro conjunto de datos conocido; De aquí viene la k. (k es un número entero y generalmente es menor que 20. Finalmente, tomamos un voto mayoritario de los k datos más similares, y la mayoría es la nueva clase que asignamos a los datos que nos pidieron clasificar. (p. 19).

Por ejemplo, si se quiere clasificar a que género pertenece una película según determinados patrones, se puede usar K-NN. En la Tabla 6 se puede observar películas, patrones como número de besos y número de patadas en una película y el tipo de película. Con lo cual, si se tiene una nueva película se podría clasificar según los mencionados patrones. (Harrington, 2012)

Tabla 6.

Películas con el número de patadas, besos y tipo por cada película

Movie title	# of kicks	# of kisses	Type of movie
California Man	3	104	Romance
He's Not Really into Dues	2	100	Romance
Beautiful Woman	1	81	Romance
Kevin Longblade	101	10	Action
Robo Slayer 3000	99	5	Action
Amped II	98	2	Action

?	18	90	Unknown
---	----	----	---------

Fuente: Harrington, P. (2012). *Machine Learning in Action*. Manning Publications Co. (p. 20)

Para poder clasificar a qué tipo de película pertenece se calcula la distancia frente a todas las películas. Para poder calcular esa distancia, se utiliza la distancia de Euclides, la cual se puede hallar con la siguiente fórmula.

$$d = \sqrt{(xA_0 - xB_0)^2 + (xA_1 - xB_1)^2} \quad (12)$$

Donde: d es la distancia, xA y xB son las coordenadas en un plano de un punto y xB son las coordenadas del otro punto. Finalmente, en la Tabla 7 se pueden observar las distancias halladas y se concluye que la nueva película pertenece al tipo de película *Action*. (Harrington, 2012)

Tabla 7.

Distancias entre cada película y la película sin tipo

Movie title	Distance to movie “?”
California Man	20.5
He’s Not Really into Dues	18.7
Beautiful Woman	19.2
Kevin Longblade	115.3
Robo Slayer 3000	117.4
Amped II	118.9

Fuente: Harrington, P. (2012). *Machine Learning in Action*. Manning Publications Co. (p. 20)

2.2.3.5.2.2 support vector machine (svm)

Si visualizamos la Figura 38 se puede observar que si trazamos una línea no se podrían separar los datos en dos grupos en ningún ejemplo. Sin embargo, en la Figura 39 sí se podría realizar esta acción. Cabe resaltar que la línea que separa los planos se llama Hiperplano de separación. Por lo cual SVM, consiste en clasificar todo lo de un lado del Hiperplano en una clase y el otro lado en otra clase. Además, es importante hablar de que los puntos más cercanos al Hiperplano de separación, llamados vectores de soporte, los cuales deberían buscar estar lo máximo posible alejados del mencionado Hiperplano, a esto se le conoce como Margen. Por lo cual, se puede concluir que el objetivo es tratar de maximizar la distancia desde el Hiperplano de separación hasta los vectores de soporte. (Harrington, 2012)

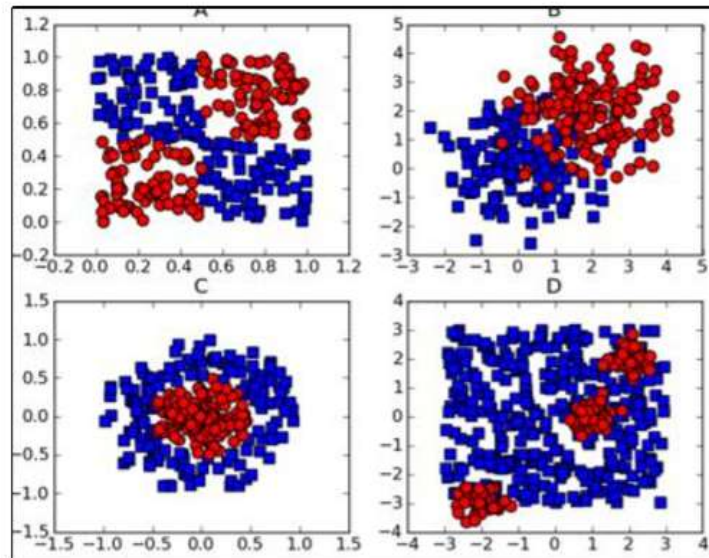


Figura 38. Datasets linealmente no separables

Fuente: Harrington, P. (2012). *Machine Learning in Action*. Manning Publications Co. (p. 102)

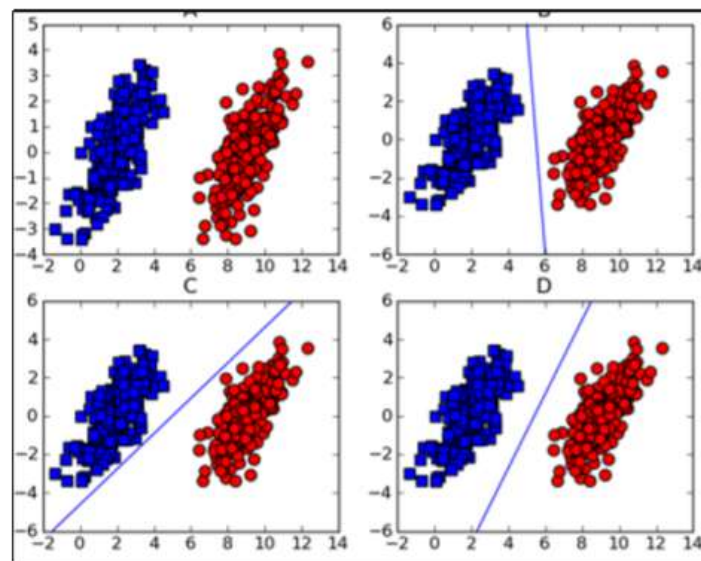


Figura 39. Datasets linealmente separables

Fuente: Harrington, P. (2012). *Machine Learning in Action*. Manning Publications Co. (p. 103)

2.2.2.2.3 redes neuronales

Haykin (2009) sostiene que “Una red neuronal artificial es un sistema masivamente paralelo y distribuido, compuesto por unidades de procesamiento simples con capacidad natural de almacenar conocimiento experimental y ponerla a disposición para uso futuro.” (p. 10)

En la Figura 40 se puede observar cómo es que se ve una Red Neuronal Artificial

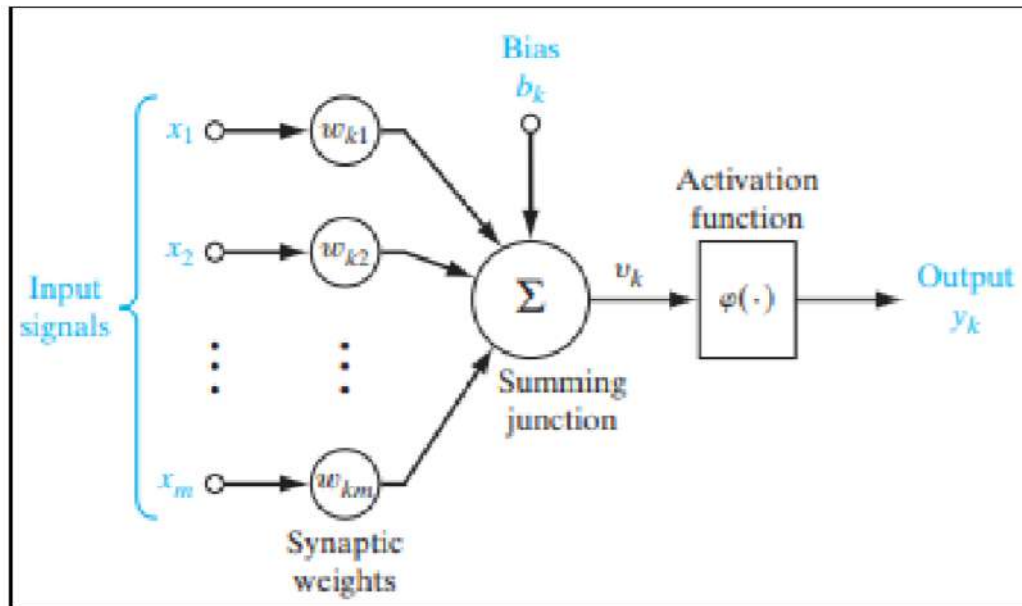


Figura 40. Modelo no lineal de una neurona

Fuente: Haykin, S. (2009). *Machine Learning in Action*. (Tercera Edición). Pearson Education, Inc. (p. 11)

2.2.2.2.3.1 perceptron

En la Figura 41 se puede observar al *Perceptron*, el cual fue desarrollado por Rosenblatt en 1958, es definido como aquel que tiene la capacidad para aprender y reconocer patrones sencillos. Asimismo, está compuesto por entradas, neurona, función de adaptación, la cual puede ser sigmoideal, tangencial, en escalón, entre otros. (Haykin, 2009)

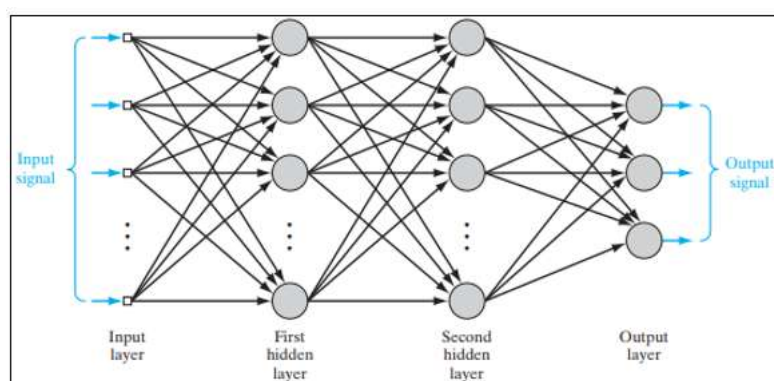


Figura 41. Gráfica del Perceptron

Fuente: Haykin, S. (2009). *Machine Learning in Action*. (Tercera Edición). Pearson Education, Inc. (p. 48)

2.2.2.2.3.2 perceptron multilayer

Haykin (2009) sostiene que la Figura 42:

Muestra el gráfico arquitectónico de un Perceptrón Multilayer con dos capas ocultas y una capa de salida. Para preparar el escenario para una descripción del perceptrón multicapa en su forma general, la red que se muestra aquí está completamente conectada. Esto significa que una neurona en cualquier capa de la red está conectada a todas las neuronas (nodos) en la anterior capa. El flujo de señal a través de la red progresa en dirección hacia adelante, de izquierda a derecha y capa por capa. (p.124)

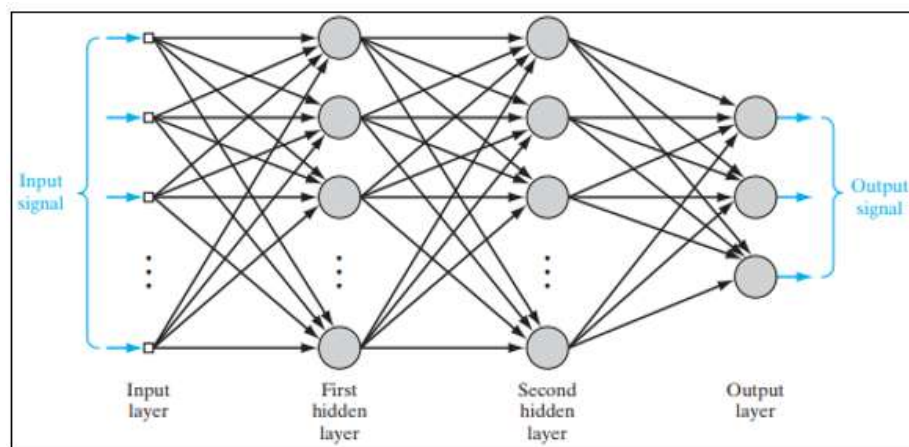


Figura 42. Arquitectura de un Perceptron Multilayer

Fuente: Haykin, S. (2009). *Machine Learning in Action*. (Tercera Edición). Pearson Education, Inc. (p. 124)

2.2.2.2.3.3 backpropagation

Ponce (2010), sostiene:

La retropropagación del error es un entrenamiento supervisado que se emplea para redes multicapa donde se ajusta el valor de los pesos en función del error generado. Esta técnica es muy empleada ya que permite tener un método de optimización que se encuentra al definir el gradiente del error y minimizarlo con respecto a los parámetros de la red neuronal. (p.225).

El algoritmo de retropropagación se explicará en los siguientes pasos:

1. Inicialmente, se tiene una red de tres capas con dos entradas y una salida, cada neurona consta de dos unidades: la suma de los productos de las entradas con sus respectivos pesos y la función de activación que viene a ser denotada por

$$y = f(e) \quad (13)$$

2. La manera de enseñarle a la red neuronal se da mediante un entrenamiento iterativo. En cada iteración los pesos de los nodos se modifican, ya que se usan nuevos datos del conjunto para realizar el entrenamiento. La propagación de las señales en las neuronas se debe a la suma de los productos de las entradas con los respectivos pesos y los resultados servirán como entradas para las neuronas de la capa oculta, para finalmente aplicarle la función de activación. Por ejemplo, en este caso, la neurona f1 es igual a: (Ponce, 2010).

$$Y = f1 (W(X1)1 * X1 + W(X2)1 * 2) \quad (14)$$

3. Lo siguiente implica comparar la salida de la red con el objetivo deseado y el resultado es llamado error de la señal. Como es imposible conocer el error de las neuronas de las capas internas, el algoritmo de backpropagation propaga el error de regreso a todas las neuronas de capas anteriores considerando los pesos de las conexiones.
4. Finalmente, habiendo calculado el error por cada neurona, se procede a modificar los pesos con la siguiente ecuación:

$$W'_{(x1)1} = W_{(x1)1} + n\alpha_1 \frac{df_1(e)}{de} x_1 \quad (15)$$

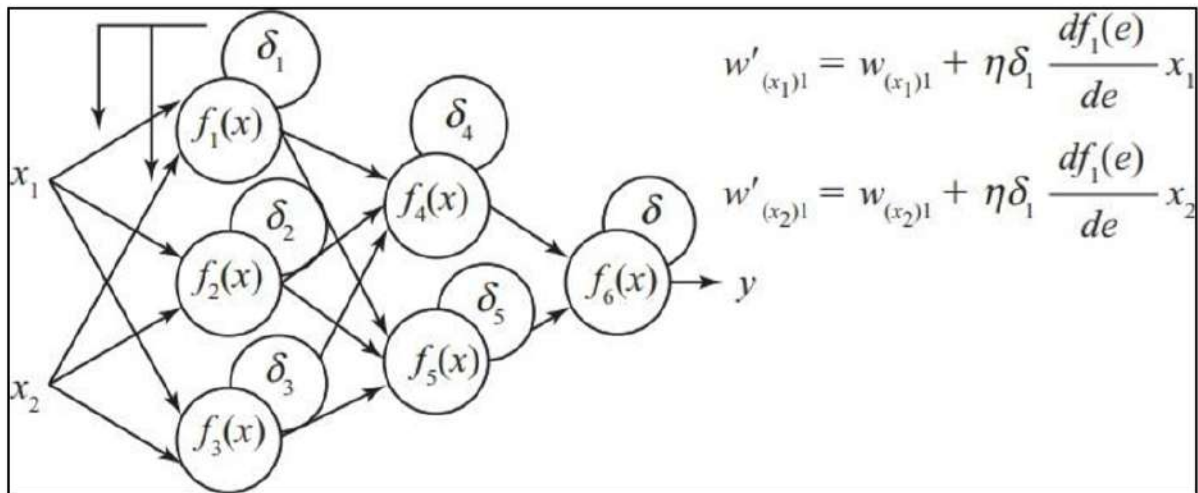


Figura 43. Actualización de pesos

Fuente: Ponce, P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. (p.228)

2.2.2.2.3.4 funciones de activación

Entre las principales funciones de activación se puede nombrar a la Función Sigmoidal, Función de *Gauss*, Función de Tangente Hiperbólica, Función Logística, entre otros.

- Función Sigmoidal:

El valor de la función es cercano a uno de los valores asintóticos, y por eso el valor de la salida está comprendido en la zona baja y alta del sigmoide. Esta función es importante recae en su derivada que siempre es positiva y cercana a cero para los valores grandes positivos o negativos y toma su valor máximo cuando $x=0$. (Ponce, 2010).

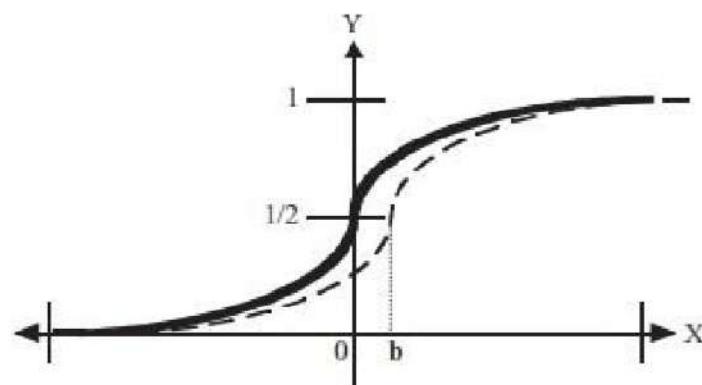


Figura 44. Función Sigmoidal

Fuente: Ponce, P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. (p.201)

$$F_K(u) = \frac{1}{1+e^{-u}} \quad (17)$$

- Función de Gauss:

Las funciones de activación tipo Gauss permiten la realización de mapeos ocultos con un solo nivel de neuronas. (Ponce, 2010).

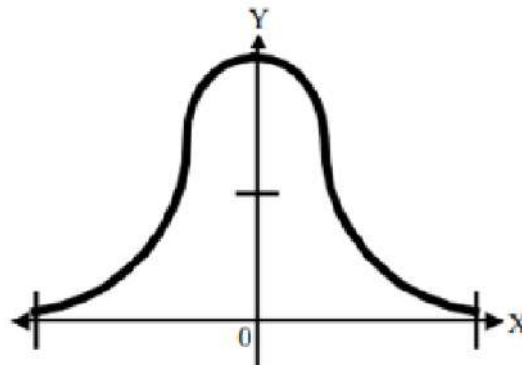


Figura 45. Función de Gauss

Fuente: Ponce, P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. (p.202)

(p.202)

$$F_K(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(\frac{x^2}{2\sigma}\right)} \quad (18)$$

- Función Escalar:

Esta función de activación está asociada a neuronas de tipo binarias. Cuando la suma de sus entradas es mayor al umbral de la neurona, la activación es 1, en caso contrario es 0 (Ponce, 2010)

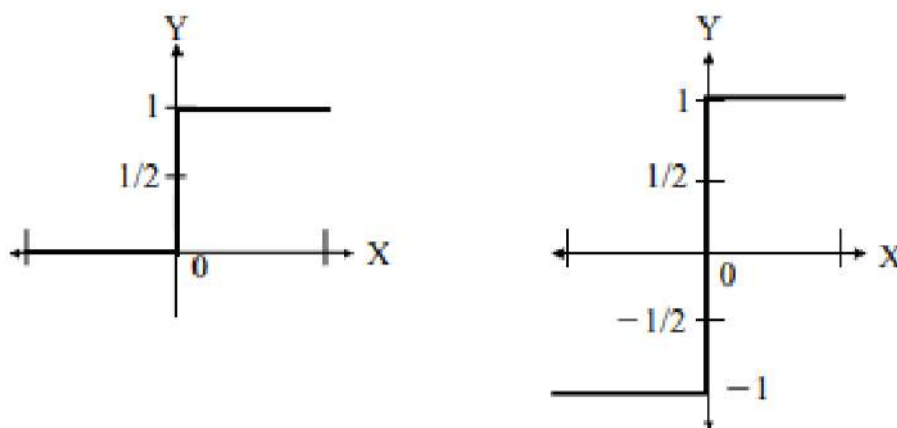


Figura 46. Función Escalar

Fuente: Ponce, P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. (p.200)

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (19)$$

$$f(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (20)$$

- Función Tangente Hipérbólica:

Esta función se aplica en los casos que presentan variaciones poco marcadas o suaves de valores positivos y negativos de la señal. Es una de las más usadas para hacer entrenamiento supervisado. Esta función toma valores de 1 y -1 (Ponce, 2010).

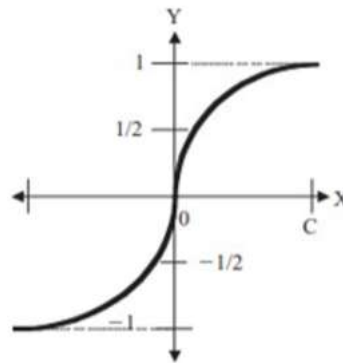


Figura 47. Función Tangente Hiperbólica

Fuente: Ponce, P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. (p.201)

(p.201)

$$F_K(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} \quad (21)$$

2.2.3.6 deep learning

Chollet (2018) sostiene que:

El aprendizaje profundo es un subcampo específico del Machine Learning: una nueva versión de las representaciones de aprendizaje a partir de datos que pone énfasis en el aprendizaje de capas sucesivas de representaciones cada vez más significativas. El Deep Learning no es una referencia a ningún tipo de comprensión más profunda lograda por el enfoque; más bien, representa esta idea de capas sucesivas de representaciones... DL a menudo implica decenas o

incluso cientos de capas sucesivas de representaciones: y todos se aprenden automáticamente de la exposición a los datos de entrenamiento. (p. 8)

En la Figura 48 se puede observar cómo es que se ve una red de *Deep Learning* entrenada.

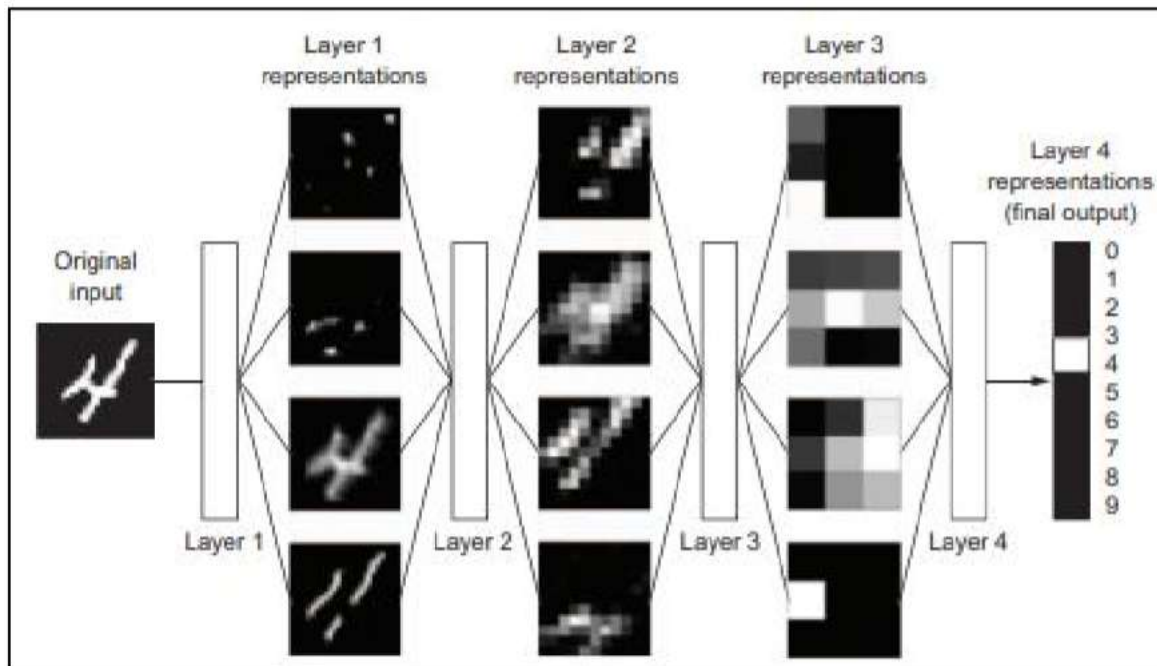


Figura 48. Representaciones de redes profundas entrenadas

Fuente: Chollet, F. (2018). *Deep Learning with Python*. Manning Publications Co. (p. 9)

Una red neuronal convolucional aprende de patrones locales. Asimismo, posee dos características muy importantes. La primera, hace referencia a la detección de patrones, ya que el autor menciona que, al reconocer la forma de una esquina en una imagen, un convnet puede reconocerla en cualquier lugar de la imagen. La segunda, hace referencia a que puede aprender jerarquías espaciales de patrones, la primera capa de convolución aprenderá sobre bordes, la segunda sobre grandes características y así sucesivamente. Por lo cual, se puede concluir que la red neuronal convolucional trabaja de manera más eficiente que una red normal, la cual tendría que entrenarse y volver a entrenarse para identificar diferentes patrones. En la Figura 26 se puede observar reflejadas las dos características mencionadas anteriormente. (Chollet, 2018)

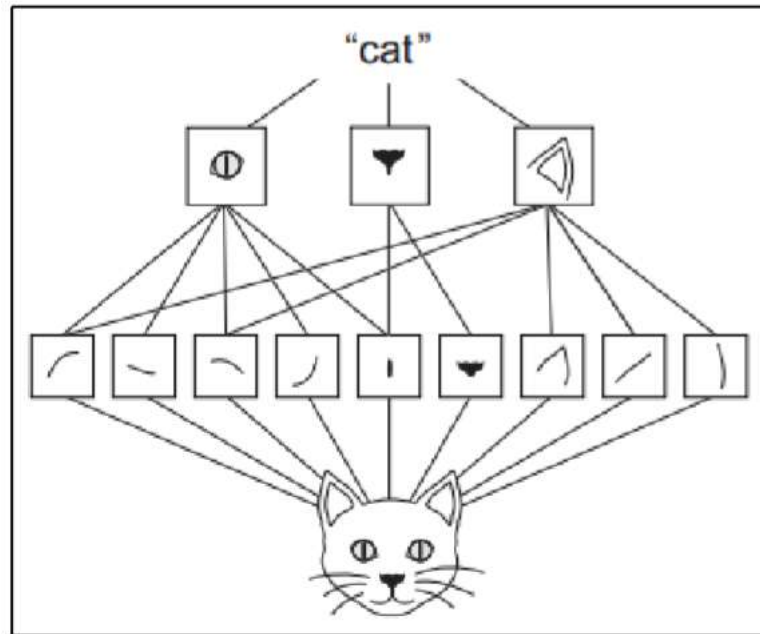


Figura 49. Características de una Red Neuronal Convolutiva

Fuente: Chollet, F. (2018). *Deep Learning with Python*. Manning Publications Co. (p. 123)

Seguidamente, en la Figura 50 se puede observar cómo es el procedimiento del funcionamiento de una convolución.

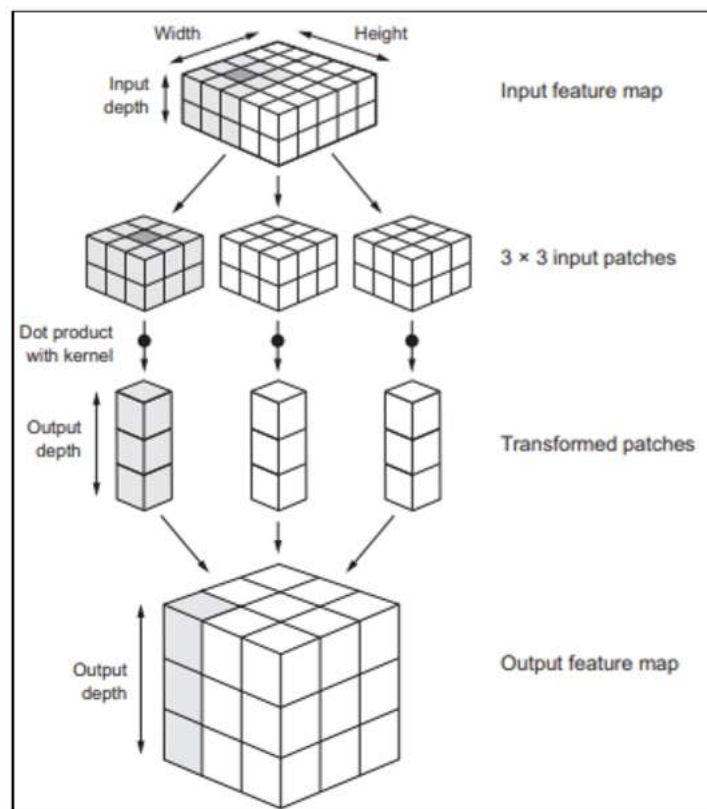


Figura 50. ¿Cómo funciona una convolución?

Fuente: Chollet, F. (2018). *Deep Learning with Python*. Manning Publications Co. (p. 125)

2.2.3.6.1 estructura:

2.2.3.6.1.1 capa de convolución

La convolución es una operación matemática, en la cual una función es aplicada por otra función. Además, la operación de convolución mayormente se denota con un asterisco. En la Ecuación 9 se puede observar la fórmula de la convolución donde x se denomina entrada y w se denomina núcleo, el índice de tiempo t solo puede tomar valores enteros y la salida es referida como mapa de características. (Goodfellow, Bengio & Courville, 2016).

$$s(t) = (x * w)(t) \quad (22)$$

Cowley (2018) sostiene que:

El proceso de convolución procede de la siguiente manera. Primero, el filtro cubre la parte superior izquierda de la matriz de la imagen. A continuación, el producto relativo al elemento del filtro se toma con la subsección de la imagen sobre la que se encuentra actualmente el filtro. Es decir, el elemento superior izquierdo del filtro se multiplica por el elemento superior izquierdo de la imagen, y así sucesivamente. Después se añaden esos resultados para producir un valor. Luego el filtro se mueve a lo largo de la imagen recorriendo una distancia que se llama zancada, y se repite el proceso. El resultado es una nueva matriz, de diferentes dimensiones en la matriz de imagen (normalmente el resultado tiene menor altura y anchura, aunque más canales). (p.2)

En la Figura 51 se puede ver como se ve una imagen original y a su derecha la imagen después de haberle aplicado un filtro de 3x3.



Figura 51. Imagen real e imagen con filtro

Fuente: Cowley, J. (2018). *Redes Neuronales Convolucionales*. (p. 3-4)

2.2.3.6.1.2. capa pooling

Una capa de *pooling*, agrupamiento, es aquella que busca reemplazar la salida de una red con una estadística resumida de salidas cercanas. Un ejemplo muy común es *Max Pooling*, la agrupación máxima, la cual reemplaza la máxima salida dentro de un vecindario rectangular. (Goodfellow, Bengio & Courville, 2016).

Cowley (2018) sostiene que:

El otro tipo de capa que se ve en las redes convolucionales es la capa de agrupación. Viene en diferentes formas, pero la que más habitualmente se utiliza es la agrupación máxima, en la que en la matriz de entrada se divide en segmentos del mismo tamaño, y se toma el valor máximo de cada segmento para rellenar el elemento correspondiente de la matriz de salida. (p.5).

En la Figura 52 se puede ver como la entrada se divide en dos cuadrantes 2x2 y se aplica *max pooling*, agrupación máxima.

$$\begin{bmatrix} 17 & 9 & 3 & 5 \\ 6 & 20 & 4 & 7 \\ 9 & 0 & 61 & 5 \\ 10 & 10 & 2 & 8 \end{bmatrix} \rightarrow \begin{bmatrix} 20 & 7 \\ 10 & 61 \end{bmatrix}$$

Figura 52. Ejemplo de Max Pooling

Fuente: Cowley, J. (2018). *Redes Neuronales Convolucionales*. (p. 5)

2.2.3.6.1.3. capa de dropout

Goodfellow, Bengio & Courville (2016) sostienen que:

Específicamente, la capa de dropout capacita al conjunto que consiste en todas las subredes que se pueden formar eliminando las unidades sin salida de la red base subyacente, como se ilustra en la Figura 53. En la mayoría de las redes neuronales modernas, basadas en una serie de transformaciones afines y no lineales, se puede eliminar efectivamente una unidad de una red multiplicando su valor de salida por cero.

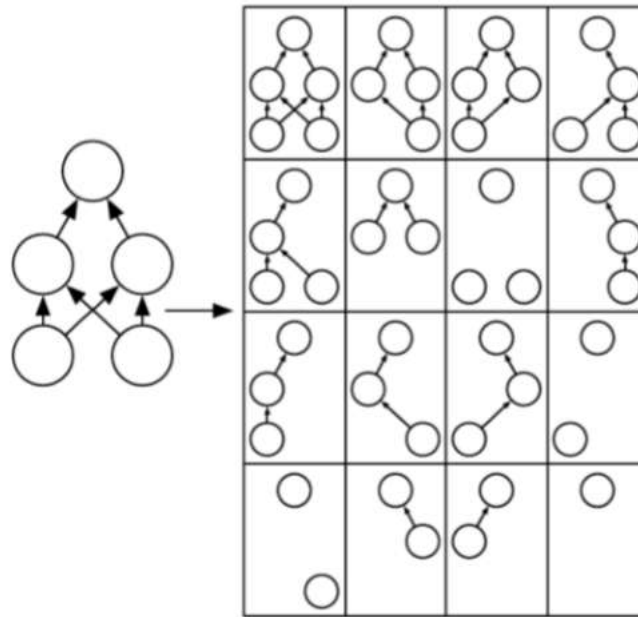


Figura 53. Ejemplo de Dropout

Fuente: Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press. (p. 258)

2.2.3.6.1.4. capa de fully connected

Cowley (2018) sostiene que:

Las redes convolucionales reales raramente se construyen solo sobre capas convolucionales. Normalmente también tienen otros tipos de capas. La más sencilla es la capa completamente conectada. Esta es tan sólo una capa de red neural normal en la que todos los resultados de la capa anterior están conectados a todos los nodos de la capa siguiente. Normalmente, esas capas van hacia el final de la red.

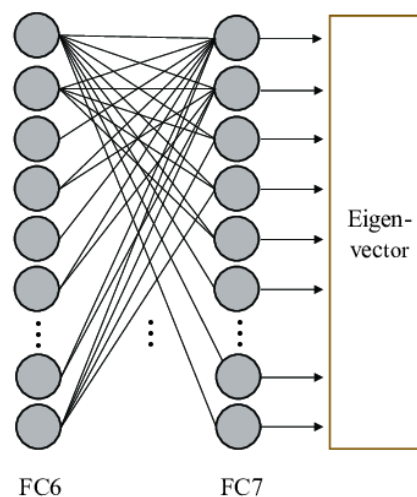


Figura 54. Ejemplo de Fully Connected

Fuente: Li, J, Tao, Q. & Wen, C. (2018). *Robust Face Recognition Using the Deep C2D-CNN Model Based on Decision-Level Fusion.* (p. 11)

2.2.3.6.2. aprendizaje de una cnn:

2.2.3.6.2.1 entrenamiento basado en gradientes

En la Figura 55 se observa una red neuronal convolucional:

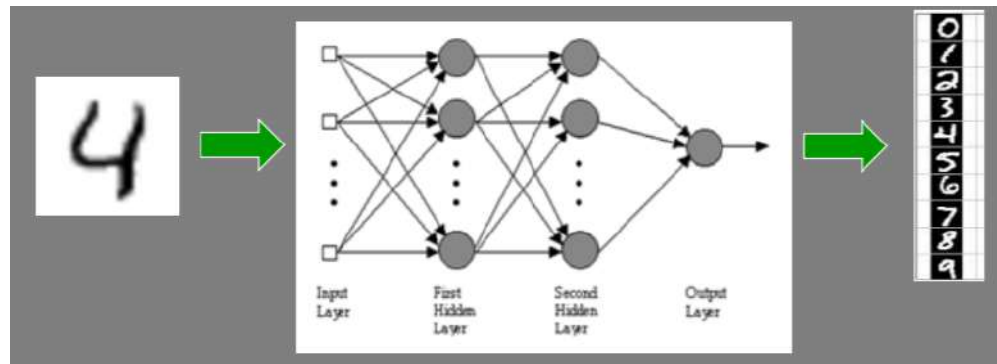


Figura 55. Ejemplo de Fully Connected

Fuente: Li, J, Tao, Q. & Wen, C. (2018). *Robust Face Recognition Using the Deep C2D-CNN Model Based on Decision-Level Fusion.* (p. 11)

$$y = f(x_0, w) \quad (23)$$

Donde:

x_0 ; es una imagen [28,28]

w ; son los parámetros de red (pesos, sesgo)

y ; salida softmax = probabilidad de que x pertenezca a una de las 10 clases 0...9

El objetivo es encontrar los parámetros w , para minimizar el error:

$$E(f(x_o, w), y_o) = -\log (f(x_o, w) - y_o) \quad (24)$$

Para ello se hace un descenso de gradiente iterativo:

$$w(t) = w(t - 1) - \lambda * \frac{-dE}{dw} (t) \quad (25)$$

Pero ¿cómo se calcula el gradiente de los pesos E wrt?

La función de pérdida E es una cadena de funciones. Se va capa por capa, desde la última capa hacia atrás, y se usa la regla de la cadena para el gradiente de funciones complejas:

$$\frac{dE}{dy_{l-1}} = \frac{dE}{dy_1} x \frac{dy_l(w, y_{l-1})}{dy_{l-1}} \quad (26)$$

$$\frac{dE}{dw_l} = \frac{dE}{dy_1} x \frac{dy_l(w, y_{l-1})}{dw_l} \quad (27)$$

2.2.3.6.2.1 backpropagation basado en gradientes

- Softmax

En este caso se considera la última capa, softmax con log-loss (ejemplo de MNIST):

$$E = -\log(p_{y0}) = -\log\left(\frac{e^{y0}}{\sum_0^9 e^{yk}}\right) = y0 + \log(\sum_0^9 e^{yk}) \quad (28)$$

Para todo $k = 0 \dots 9$, excepto $k0$ (respuesta correcta) queremos disminuir p_k :

$$\frac{dE}{dy_k} = \frac{e^{yk}}{\sum_0^9 e^{yk}} = p_k, \quad (29)$$

para $k = k0$ (respuesta correcta) se quiere aumentar p_k :

$$\frac{dE}{dy_{k0}} = -(1 - p_{k0}) \quad (30)$$

- Fully Connected layer

La capa *Fully Connected* es simplemente una multiplicación de Matrix - Vector:

$$y_l = W_l * y_{l-1} \quad (31)$$

Así que:

$$\frac{dE}{dy_{l-1}} = \frac{dE}{dy_l} * W_l^T \quad (32)$$

y:

$$\frac{dE}{dw_l} = \frac{dE}{dy_l} * y_{l-1} \quad (33)$$

Nota: Se necesita y_{l-1} , por lo que se debe mantenerlos en pase hacia adelante.

- Pooling

Unidad lineal rectificada:

$$y_l = \max(0, y_{l-1})$$

Así que:

$$\frac{dE}{dy_{l-1}} = \begin{cases} = 0, & \text{if } (y_l < 0) \\ = \frac{dL}{dy_l}, & \text{otherwise} \end{cases} \quad (34)$$

- *ReLU*

Forward:

for ($p = 0; p < k; p++$)

for ($q = 0; q < k; q++$)

$$y_n(x, y) = \max(y_n(x, y), y_{n-1}(x + p, y + q)); \quad (35)$$

Backward:

$$\frac{dL}{dy_{n-1}}(x + p, y + q) = \begin{cases} = 0, & \text{if } (y_n(x, y) \neq y_{n-1}(x + p, y + q)) \\ = \frac{dL}{dy_n}(x, y), & \text{otherwise} \end{cases}$$

- *Convolutonal layer*

Se hace uso de la regla de la cadena para cada capa convolucional:

Gradiente $\frac{dE}{dy_{l-1}}$ es la suma de la convolución con gradientes $\frac{dE}{dy_l}$ en general mapas de características de la capa "superior":

$$\frac{dE}{dy_{l-1}} = \frac{dE}{dy_l} x \frac{dy_l(w, y_{l-1})}{dy_{l-1}} = \sum_{n=1}^N \text{back_corr}(W, \frac{dE}{dy_l}) \quad (36)$$

El gradiente de E wrt w es la suma de todos los "píxeles" (x, y) en la entrada mapa:

$$\frac{dE}{dw_l} = \frac{dE}{dl} x \frac{dy_l(w, y_{l-1})}{dw_l} = \sum_{\substack{0 \leq x \leq X \\ 0 < y < Y}} \left(\frac{dE}{dy_l}(x, y) \circ y_{l-1}(x, y) \right) \quad (37)$$

2.2.3.6.3. arquitecturas:

2.2.3.6.3.1 vgg

La arquitectura de esta red fue desarrollada por Simonyan y Zisserman. VGGNet consta de 16 capas convolucionales, presenta solo convoluciones 3x3, pero muchos filtros. Entre sus principales usos están la extracción de características de las imágenes. La configuración de

pesos de VGGNet se encuentra disponible para cualquier usuario y ha sido usada en desafíos como un extractor de características de línea de base. Cabe indicar, que VGGNet consta de 138 millones de parámetros. (Simonyan y Zisserman, 2015)

Simonyan y Zisserman (2015) sostienen que:

El conjunto de datos ImageNet contiene imágenes de tamaño fijo de 224 * 224 y tienen canales RGB. Entonces, se tiene un tensor de (224, 224, 3) como entrada. Este modelo procesa la imagen de entrada y genera un vector de 1000 valores.

$$\hat{y} = \begin{bmatrix} \widehat{y_0} \\ \widehat{y_1} \\ \widehat{y_2} \\ \widehat{y_3} \\ \vdots \\ \vdots \\ \vdots \\ \widehat{y_{999}} \end{bmatrix}$$

Este vector representa la probabilidad de clasificación para la clase correspondiente. Suponga que se tiene un modelo que predice que la imagen pertenece a la clase 0 con probabilidad 1, clase 1 con probabilidad 0.05, clase 2 con probabilidad 0.05, clase 3 con probabilidad 0.03, clase 780 con probabilidad 0.72, clase 999 con probabilidad 0.05 y todas las demás clases con 0. Entonces, el vector de clasificación para esto será:

$$\hat{y} = \begin{bmatrix} \widehat{y_0} = 0.1 \\ 0.05 \\ 0.05 \\ 0.03 \\ \vdots \\ \vdots \\ \vdots \\ \widehat{y_{780}} = 0.72 \\ \vdots \\ \vdots \\ \widehat{y_{999}} = 0.05 \end{bmatrix}$$

Para asegurarse de que estas probabilidades sumen 1, se usa la función softmax, la cual se define como:

$$P(y = j | \theta^{(i)}) = \frac{e^{\theta^{(i)}}}{\sum_{k=0}^K e^{\theta_k^{(i)}}} \quad (38)$$

Donde:

$$\theta = w_0x_0 + w_1x_1 + \dots + w_kx_k = \sum_{i=0}^k w_ix_i = w^T x \quad (39)$$

Después, se toma los 5 candidatos más probables en el vector:

$$C = \begin{bmatrix} 780 \\ 0 \\ 1 \\ 2 \\ 999 \end{bmatrix}$$

y el vector fundamental se define de la siguiente manera:

$$G = \begin{bmatrix} G_0 \\ G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} 780 \\ 2 \\ 999 \end{bmatrix}$$

Luego se define la función Error de la siguiente manera:

$$E = \frac{1}{n} \sum_k \min_i d(c_i, G_k) \quad (40)$$

Donde:

$$d = 0 \text{ if } c_i = G_k \text{ else } d = 1$$

Por lo tanto, la función de pérdida para este ejemplo es:

$$E = \frac{1}{3} (\min_i d(c_i, G_1) + \min_i d(c_i, G_2) + \min_i d(c_i, G_3)) \quad (41)$$

$$E = \frac{1}{3} (0 + 0 + 0)$$

$$E = 0$$

dado que, todas las categorías en el vector fundamental están en la matriz de los 5 principales predichos, la pérdida se convierte en 0.

Arquitectura:

La entrada a la red es imagen de dimensiones (224, 224, 3). Las dos primeras capas tienen 64 canales de tamaño de filtro 3 * 3 y el mismo relleno. Luego, después de max pooling de (2, 2), dos capas que tienen capas de convolución de tamaño de filtro 256 y tamaño de filtro (3, 3). Esto seguido de max pooling de (2, 2) que es la misma que la capa anterior. Luego hay 2 capas de convolución de tamaño de filtro (3, 3) y filtro 256.

Después de eso, hay 2 conjuntos de 3 capas de convolución y un max pooling. Cada uno tiene 512 filtros de tamaño (3, 3) con el mismo relleno. Esta imagen se pasa a la pila de dos capas de convolución. En estas capas de convolución max pooling, los filtros que se usan son del tamaño $3 * 3$ en lugar de $11 * 11$ en AlexNet y $7 * 7$ en ZF-Net. En algunas de las capas, también se utiliza $1 * 1$ píxel para manipular el número de canales de entrada. Hay un relleno de 1 píxel (el mismo relleno) realizado después de cada capa de convolución.

Después de la pila de convolución y max pooling, se obtiene un mapa de características (7, 7, 512). Se aplana esta salida para ser convertida en un vector de características (1, 25088). Después de esto, hay 3 capas fully connected, la primera capa toma la entrada del último vector de características y genera un vector (1, 4096), la segunda capa también genera un vector de tamaño (1, 4096), pero la tercera capa genera 1000 canales para 1000 clases de desafío ILSVRC. Después, la salida de la tercera capa fully connected pasa a la capa softmax para normalizar el vector de clasificación. Luego, de la salida del vector de clasificación, las 5 categorías principales pasan para la evaluación. Todas las capas ocultas utilizan ReLU como función de activación. ReLU es más eficiente computacionalmente porque da como resultado un aprendizaje más rápido y también disminuye la probabilidad de que desaparezca el problema del gradiente.

Configuración:

En la Figura 56 se enumeran diferentes arquitecturas VGG. Además, se puede ver que existen 2 versiones de VGG-16 (C y D). No hay mucha diferencia entre ellas, excepto por la primera que posee una capa de convolución (3, 3) y en la segunda la convolución del tamaño del filtro es de (1, 1). Estas dos contienen 134 millones y 138 millones de parámetros respectivamente. Por último, la diferencia con VGG-19 (E) son las 19 capas convolucionales a diferencia de las 16 capas de VGG-16 y los 144 millones de parámetros.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 56. Arquitectura de VGG16

Fuente: Simonyan, K. & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. (p. 3)

2.2.3.6.3.2 alexnet

La red tiene una arquitectura que consiste en 11x11, 5x5, 3x3, convoluciones, max pooling, dropout, aumento de datos, activaciones de ReLU, SGD con impulso. Adjunta activaciones ReLU después de cada capa convolucional y fully connected. AlexNet recibió capacitación durante 6 días simultáneamente en dos GPU Nvidia Geforce GTX 580, razón por la cual su red se divide en dos canales. AlexNet fue diseñado por el grupo SuperVision, formado por Alex Krizhevsky, Geoffrey Hinton e Ilya Sutskever. (Krizhevsky, Sutskever y Hinton, 2012)

En la Figura 57 se puede observar la arquitectura de la red.

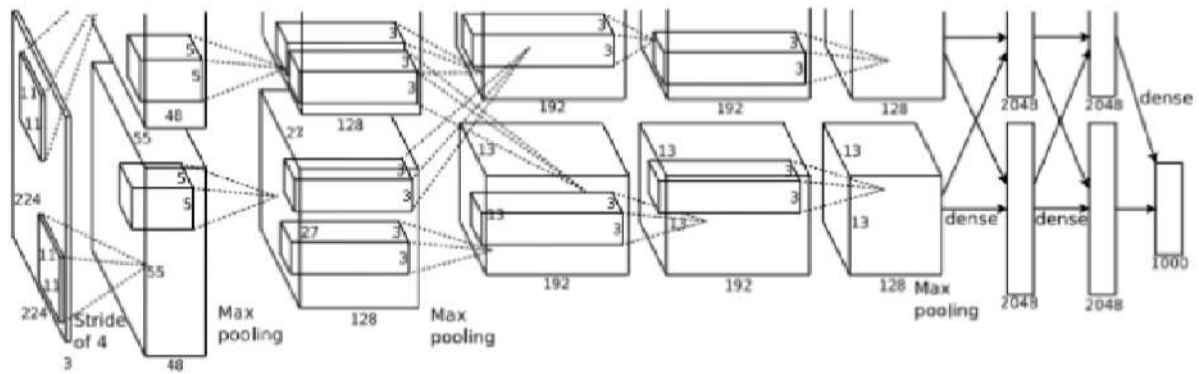


Figura 57. Arquitectura de AlexNet

Fuente: Krizhevsky, A., Sutskever, I. & Hinton, G. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. (p. 5)

2.2.3.6.3.3 googlenet

La arquitectura de esta red fue inspirada en LeNet, la mencionada arquitectura adicionó un elemento nuevo módulo de inicio. Este módulo se basa en varias convoluciones muy pequeñas para reducir drásticamente el número de parámetros. Además, esta arquitectura está compuesta por 22 capas en profundidad y el número de parámetros son reducidos desde 60 millones (AlexNet) a 4 millones. (Szegedy et al. 2014)

En la Figura 58 se puede observar la arquitectura de la red.

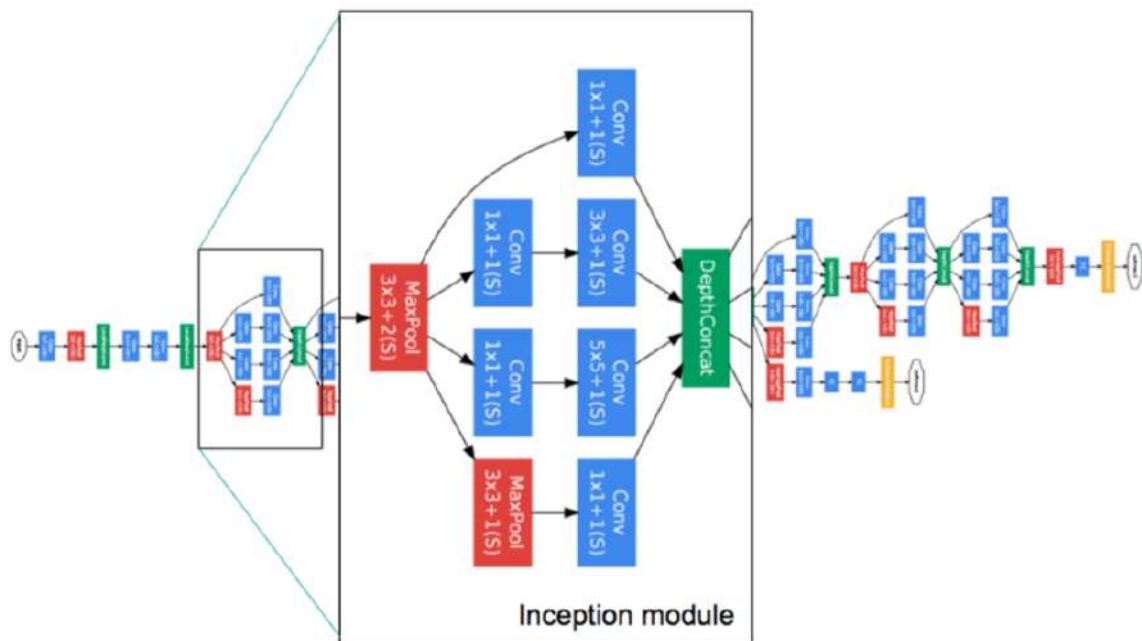


Figura 58. Arquitectura de GoogLeNet

Adaptado de: Szegedy et al. (2014). *Going deeper with convolutions*. (p. 6)

2.2.3.6.3.4 resnet

He, Zhang, Ren y Sun (2015) sostienen que:

Después de la primera arquitectura basada en CNN (AlexNet) que ganó la competencia ImageNet 2012, cada arquitectura ganadora posterior utiliza más capas en una red neuronal profunda para reducir la tasa de error. Esto funciona para una menor cantidad de capas, pero cuando se aumenta la cantidad de capas, existe un problema común en el aprendizaje profundo asociado con el llamado gradiente de desaparición / explosión. Esto hace que el gradiente sea 0 o demasiado grande. Por lo tanto, cuando se aumenta el número de capas, también aumenta la tasa de error de entrenamiento y prueba.

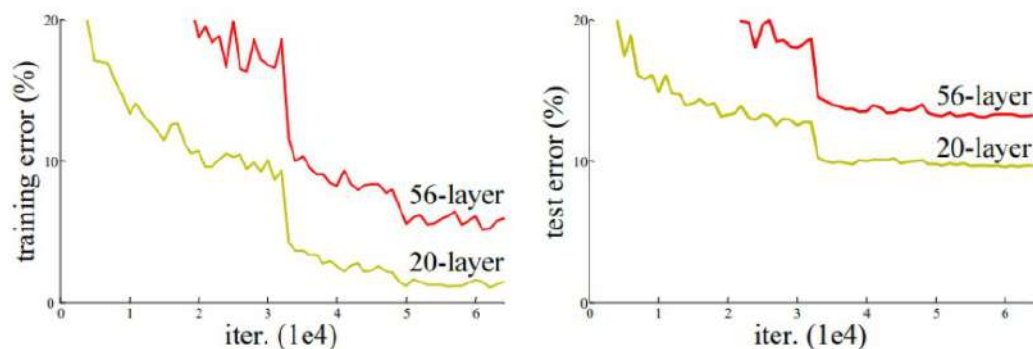


Figura 59. Error de entrenamiento (izquierda) y error de prueba (derecha) en CIFAR-10 con redes “planas” de 20 y 56 capas.

Fuente: He, K., Zhang, X., Ren, S. & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. (p.1)

En la Figura 59 se puede observar que una CNN de 56 capas da más tasa de error, tanto en el conjunto de datos de entrenamiento como de prueba, que una arquitectura CNN de 20 capas. Si esto fue el resultado de un ajuste excesivo, entonces se debe tener un error de entrenamiento menor en la CNN de 56 capas, pero también tiene un mayor error de entrenamiento. Después de un análisis sobre la tasa de error, los autores pudieron llegar a la conclusión de que se debe a un gradiente de desaparición / explosión.

Bloque residual:

Para resolver el problema del gradiente de desaparición / explosión, esta arquitectura introdujo el concepto llamado Red residual. En esta red se usa una técnica llamada salto

de conexiones. El salto de conexiones omite el entrenamiento de algunas capas y se conecta directamente a la salida.

El enfoque detrás de esta red es, en lugar de capas, aprender el mapeo subyacente, permite que la red se ajuste al mapeo residual. Entonces, en lugar de decir $H(x)$, mapeo inicial, deje que la red se ajuste.

$$F(x) := H(x) - x \quad (42)$$

Lo que da como resultado:

$$H(x) := F(x) + x \quad (43)$$

La ventaja de agregar este tipo de conexión de omisión es que, si alguna capa daña el rendimiento de la arquitectura, la regularización la omitirá. Entonces, esto da como resultado el entrenamiento de una red neuronal muy profunda sin los problemas causados por el gradiente de desaparición / explosión. Los autores del artículo experimentaron con 100-1000 capas en el conjunto de datos CIFAR-10.

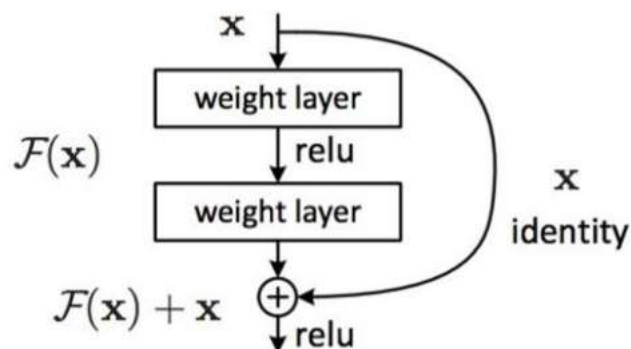


Figura 60. Conexiones saltadas

Fuente: He et al., (2015). *Deep Residual Learning for Image Recognition*. (p.2)

Existe un enfoque similar llamado "redes de carreteras", estas redes también utilizan conexión de salto. Al igual que en LSTM, estas conexiones de salto también utilizan puertas paramétricas. Estas puertas determinan cuánta información pasa a través de la conexión de salto. Sin embargo, esta arquitectura no ha proporcionado una precisión mejor que la arquitectura ResNet.

En la Figura 61 se puede observar la arquitectura de la red.

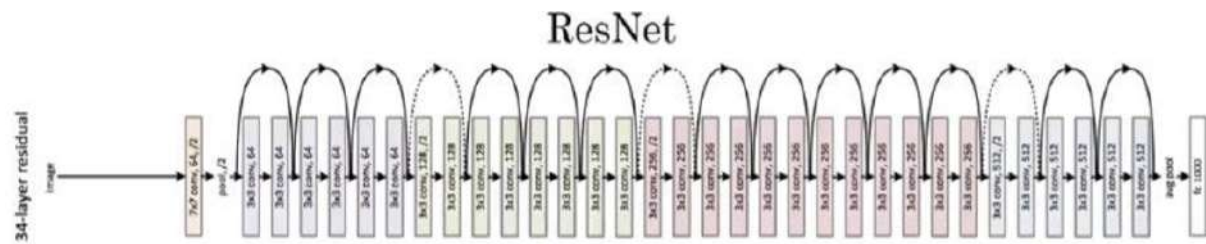


Figura 61. Arquitectura de ResNet

Fuente: He et al., (2015). *Deep Residual Learning for Image Recognition*. (p.4)

2.3 Marco Conceptual o Contexto de la Investigación

2.3.1 inteligencia artificial

La inteligencia artificial es una rama que está incluida dentro de las ciencias computacionales que busca emular la inteligencia humana a través de algoritmos; máquinas y así realizar las actividades de una manera más eficiente que el ser humano. (Russell y Norving, 2010)

2.3.2 machine learning

Machine Learning es una rama de las ciencias computacionales que usa ingeniería y estadística para solucionar problemas que implican la capacidad de aprendizaje de una máquina. (Harrington, 2012)

2.3.3 aprendizaje supervisado

Aprendizaje Supervisado consiste en permite buscar características en común entre las variables predictoras y los datos históricos relacionándolos con una variable objetivo. (Harrington, 2012)

2.3.4 support vector machine

Es un clasificador que es considerado como uno de los mejores con respecto al stock cuyas ventajas son un error de clasificación pequeño y con una fácil interpretación. (Harrington, 2012)

2.3.5 redes neuronales

Son un algoritmo que se basa en los principios de las redes biológicas para poder aprender y así predecir valores en base a patrones aprendidos. (Ponce, 2010)

2.3.6 regresión logística

En estadística, la regresión logística es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de las variables independientes o

predictoras. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores. (De la Fuente, 2011)

2.3.7 árbol de decisiones

Es un algoritmo de aprendizaje supervisado que es mayormente usado para problemas de clasificación. Sorprendentemente funciona para variables dependientes categóricas y continuas. En este algoritmo se separa la base de datos en 2 partes homogéneas. Esto se realiza basándose en la mayoría de los atributos significantes para crear grupos con la mayor cantidad de diferencias posibles. (González, L. 2018)

2.3.8 random forest

Random forest (o random forests) también conocidos en castellano como "Bosques Aleatorios" es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. (Donges, N.)

2.3.9 trastorno

“Cambio o alteración que se produce en la esencia o las características permanentes que conforman una cosa o en el desarrollo normal de un proceso.” (Diccionario Google, 2018)

2.3.10 degeneración

“Proceso por el que una persona o una cosa pasa a un estado peor del original por perder progresivamente cualidades que tenían.” (Diccionario Google, 2018)

2.3.11 síntoma

Según RANM (2012) es la “Manifestación de una enfermedad o de un síndrome que solo es percibida por el individuo que lo padece. Cuando una alteración puede ser percibida tanto por el enfermo como por un observador externo es un signo (por ejemplo, la fiebre), pero la sensación subjetiva que la acompaña (por ejemplo, la cefalea) es un síntoma”.

2.3.12 neurotransmisor

Según EnciclopediaSalud “Sustancia química segregada por las neuronas en las sinapsis. Los NT se liberan en la sinapsis con la llegada de un impulso nervioso, traspasan el espacio sináptico y estimulan la neurona receptora. Principales neurotransmisores: noradrenalina, dopamina, serotonina, acetilcolina” (EnciclopediaSalud)

2.3.13 dopamina

Según EnciclopediaSalud “Neurotransmisor del grupo de las catecolaminas. Las neuronas dopaminérgicas se concentran en el mesencéfalo, sustancia negra y área tegmental ventral. En el cerebro humano hay tres grandes sistemas dopaminérgicos: el nigro-estriado, el tubero-infundibular y el mesolímbico-mesocortical.” (EnciclopediaSalud)

2.4 Hipótesis

Para la identificación de la hipótesis general y de las hipótesis específicas se determinó de acuerdo a los problemas y objetivos identificados previamente. Asimismo, las hipótesis han sido identificadas en base a una realidad actual que afecta a una gran parte de la población mundial que es la enfermedad de Parkinson.

2.4.1 hipótesis general

HP: Mediante la implementación de un modelo de visión computacional se logra determinar el prediagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona.

2.4.2 hipótesis específicas

HE1: La identificación y creación de bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson permite el desarrollo del modelo.

HE2: La implementación de técnicas de procesamiento de imágenes a imágenes de manuscritos influye positivamente para poder aumentar la calidad de la imagen.

HE3: La implementación de técnicas de visión computacional favorece para poder extraer vectores característicos de patrones como color, forma o textura de las imágenes.

HE4: La utilización de técnicas de *machine learning* y *deep learning* influyen positivamente en la clasificación de la imagen de la escritura de una persona respecto a si pertenece a una persona con prediagnóstico de la enfermedad de Parkinson.

HE5: La utilización de métricas influye positivamente en la evaluación de la imagen de la escritura de una persona respecto a si pertenece a una persona con prediagnóstico de la enfermedad de Parkinson.

Capítulo 3: Metodología

En el presente capítulo se define el diseño, tipo y enfoque de la investigación. Asimismo, se explica la población y muestra, es decir las bases de datos que van a ser usadas y cuantos datos se tienen para el desarrollo del algoritmo predictivo. Finalmente, se explican las técnicas usadas para la recolección de datos, el preprocesamiento de los mismos y los modelos de *Machine Learning* y *Deep Learning* para la creación del modelo de clasificación.

3.1 Diseño, tipo y enfoque de Investigación

3.1.1 diseño

El diseño del presente trabajo de investigación es de tipo experimental, debido a que se van a analizar las variables y la relación que existe entre ellas. Asimismo, dado que el presente trabajo tiene como etapa inicial el análisis de imágenes, estas tienen que ser procesadas con técnicas de visión computacional para poder extraer un vector característico, el cual será la entrada para el siguiente paso el cual es aplicar modelos de *machine learning* y *deep learning*. (Hernández Sampieri, R., Fenández Collado, C., & Baptista Lucio, P., 2010).

3.1.1 tipo – nivel

El nivel del presente trabajo de investigación es explicativo, debido a que el diseño de un sistema de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona busca determinar cómo se puede saber si una persona presenta la enfermedad de Parkinson a través de su escritura, por lo cual, para ello, se establece una relación tipo causa efecto, es decir si se conoce o se encuentra un determinado patrón en la escritura de una persona con Parkinson se podría determinar si padece la enfermedad o no. (Hernández Sampieri et al., 2010).

3.1.2 enfoque

El enfoque de la presente investigación es cuantitativo, debido a que se utilizan instrumentos para la identificación y medición del objeto de estudio el cual es el pre-diagnóstico de la enfermedad de Parkinson en una persona a través de la escritura, se utilizarán determinadas técnicas de visión computacional cuyos resultados numéricos, los cuales provienen de análisis estadísticos, servirán como entrada para poder implementar los modelos de *machine learning* y *deep learning*. (Hernández Sampieri et al., 2010).

3.2 Población y Muestra

Tabla 8.

Tabla de Población y Muestra

Población	Imágenes de formas de espirales y meandros de escritura de personas, extraídas de la Facultad de Medicina Botucatu, Universidad Estatal de São Paulo, Brasil en el año 2016 y de Activate Arequipa Liga contra el Parkinson, Perú en el año 2020
Muestra	<p>632 imágenes etiquetadas respectivamente para meandros y espirales,</p> <p>de la escritura de personas extraídas de la Facultad de Medicina Botucatu, Universidad Estatal de São Paulo, Brasil en el año 2016.</p> <p>1080 imágenes etiquetadas para meandros y 1296 etiquetadas para espirales, de la escritura de personas, extraídas de Activate Arequipa Liga contra el Parkinson, Perú en el año 2020.</p> <p>Muestreo no probabilístico o dirigido, debido a que se seleccionará ciertas escrituras de personas con un propósito, el cual es determinar mediante dicha escritura si la persona padece de la enfermedad de Parkinson.</p> <p>De la población interesa seleccionar personas saludables y personas que padecen la enfermedad de Parkinson lo cual sirve para la construcción del modelo. Sin embargo, en el caso de la base de datos proveniente de Brasil se encuentra un desbalance de datos que existe entre personas saludables y las que no son, por lo cual, se completó con imágenes hechas manualmente con la misma prueba, pero personas sanas.</p> <p>Adicionalmente, cabe mencionar que esta muestra al ser aplicada en un modelo de <i>machine learning</i> posteriormente va a ser particionada en un <i>train</i> (70%), <i>test</i> (20%) y un <i>val</i> (10%). Con el fin de poder entrenar el modelo (<i>train</i>), probarlo (<i>test</i>) y validarlo con (<i>val</i>). Finalmente, los modelos y resultados serán evaluados con validación cruzada con el fin de evitar sobreajuste.</p>
Unidad de Análisis	Una imagen de espiral o meandro proveniente de la escritura de una persona

Variable (s) y tipo (s) de variable (s)	Variable cuantitativa y discreta, debido a que el presente trabajo de investigación está centrado en variables numéricas y la precisión o exactitud.
---	--

Fuente: Elaboración propia

3.3 Operacionalización de Variables

Tabla 9.

Matriz de variables principales

DEFINICIÓN DE VARIABLES		
VARIABLE Y DEFINICIÓN	INDICADOR	FÓRMULA DE INDICADOR
ENFERMEDAD DE PARKINSON Enfermedad crónica y degenerativa del sistema nervioso que se caracteriza por falta de coordinación y rigidez muscular y temblores	Escritura de una persona	$\frac{\text{Cantidad de personas con pre – diagnóstico de EP}}{\text{Cantidad de personas con EP}}$
VISIÓN COMPUTACIONAL Disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real	Resolución de la imagen	Cantidad de píxeles en la imagen
	Preprocesamiento de la imagen	Nivel de nitidez, contraste o ruido de la imagen
MODELO DE CLASIFICACIÓN	Accuracy	$\frac{TP + TN}{TP + FP + FN + TN}$
	Precision	$\frac{TP}{TP + FP}$

Modelo con la capacidad de clasificar, según la clase de la variable, evaluado con las métricas más conocidas y robustas	Recall	$\frac{TP}{TP + FN}$
	F1	$2 * \frac{Precision * Recall}{Precision + Recall}$
	ROC AUC	$P(score(x^+) > score(x^-))$

Fuente: Elaboración propia

Donde:

TP = True Positives

FP = False Positives

FN = False Negatives

TN = True Negatives

3.4 Técnicas de Recolección de Datos

Lo que se busca analizar en este punto es mostrar cuál fue el examen de escritura que se le aplicó a los miembros de la Asociación Actívate Liga contra el Parkinson Arequipa y personas fuera de la Asociación para recolectar las imágenes de su manuscrito. Es importante resaltar que existen dos exámenes de escritura, la Figura 4 muestra el examen que se aplicó en Brasil y las Figura 62 (primera hoja) y Figura 63 (segunda hoja) muestra el examen que se aplicó en Perú.

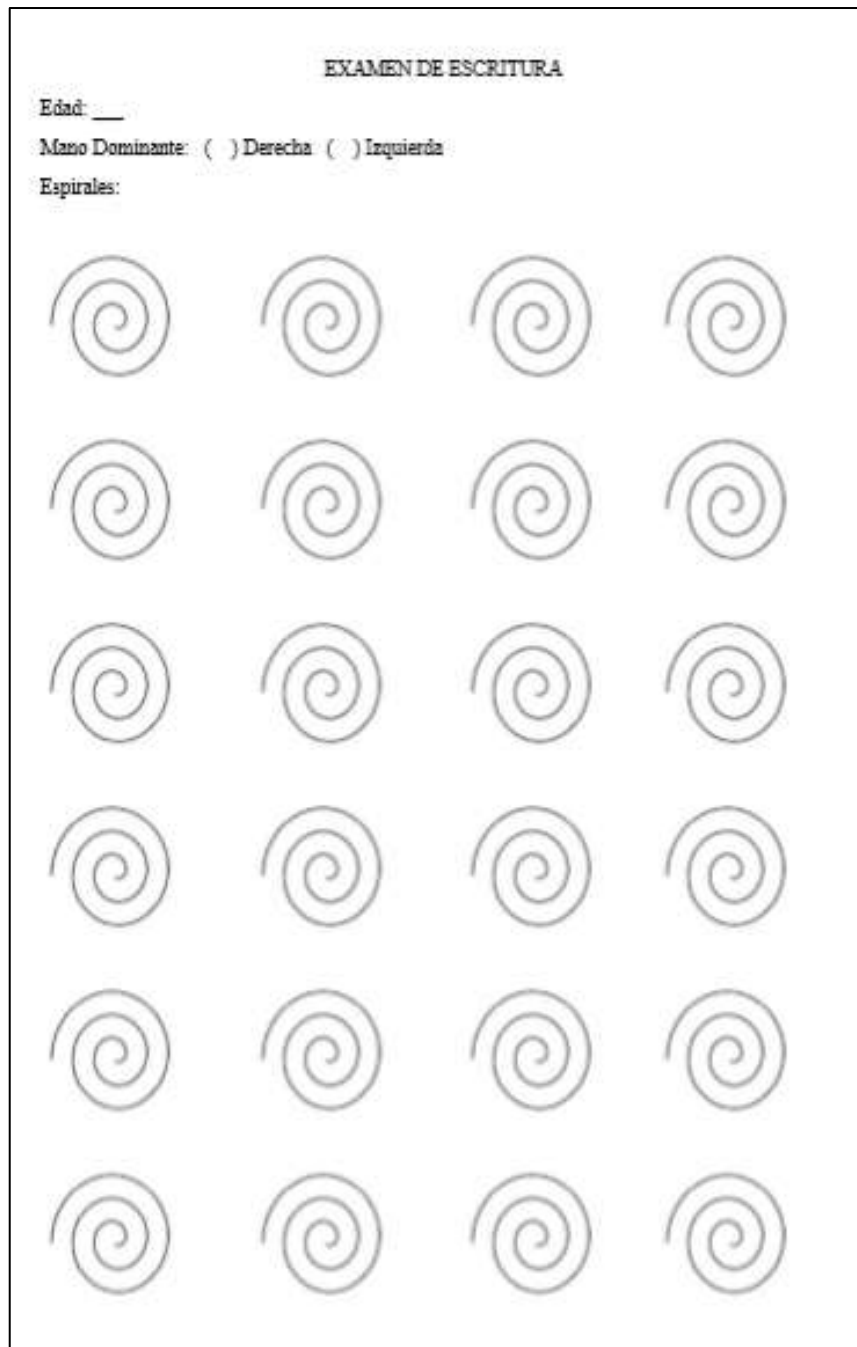


Figura 62. Primera hoja del examen de escritura

Fuente: Elaboración propia

EXAMEN DE ESCRITURA

Edad: ____

Mano Dominante: () Derecha () Izquierda

Meandros:









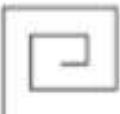















			
			
			
			
			
			

Figura 63. Primera hoja del examen de escritura

Fuente: Elaboración propia

El Examen de Escritura fue validado por una doctora experta en Neurología como se observa en el Anexo 2.

3.5 Técnicas para el Procesamiento y Análisis de la Información

3.5.1 metodología de la implementación de la solución

Según Szeliski R. (2010), en la Visión Computacional existen diferentes etapas, las cuales consisten en el desarrollo desde cómo se capta una imagen hasta la interpretación de esta. Para llegar a la etapa final la imagen captada debe seguir un proceso el cual se va a explicar detalladamente en los siguientes puntos. La metodología del presente trabajo de investigación se puede visualizar de manera gráfica en la Figura 64.

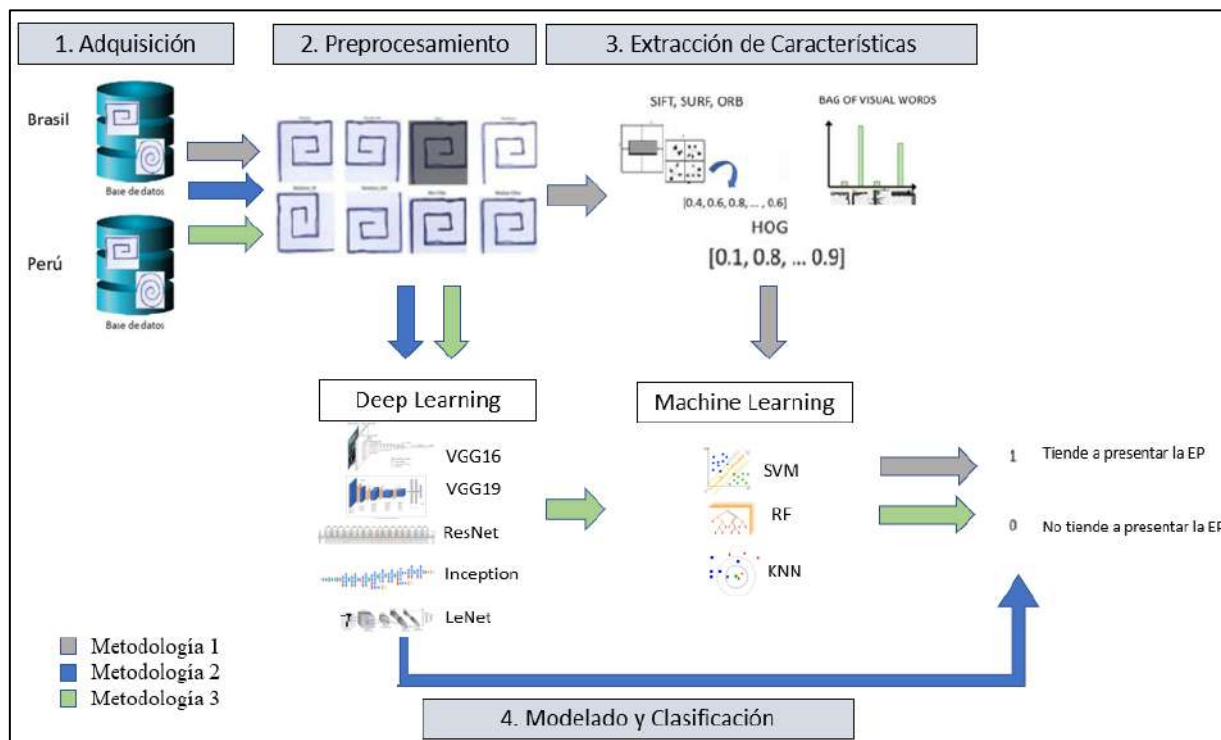


Figura 64. Metodología de la investigación

Fuente: Elaboración propia

3.5.1.1 adquisición:

En esta etapa se describe la adquisición de los dos *datasets* con los cuales se trabajó en el presente trabajo de investigación. A continuación, se presentará una tabla con las tareas y actividades donde se detallará esta etapa de adquisición:

Tabla 10.

Actividades etapa de Adquisición

Actividades	Descripción	Tareas
-------------	-------------	--------

1.- Identificar bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson.	Identificación de bases de datos que contengan trazos de personas diagnosticadas con la EP y personas sanas.	<p>Analizar e identificar bases de datos que contengan trazos de personas con EP y personas sanas.</p> <p>Validar que la base de datos esté respaldada por médicos o una entidad médica.</p> <p>Descargar la base de datos del repositorio en el que se encuentra.</p>
2.- Crear una base de dato de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson.	Elaboración de una base de datos que contenga trazos de personas diagnosticadas con la EP y personas sanas.	<p>Identificar un conjunto de personas que presenten la EP y personas sanas.</p> <p>Contactarse con la entidad responsable de las personas identificadas.</p> <p>Crear el examen de escritura.</p> <p>Validar el examen de escritura por un especialista en EP.</p> <p>Solicitar permisos para poder aplicar un examen de escritura a las personas identificadas.</p> <p>Aplicar el examen de escritura a las personas que presenten la EP y a personas sanas.</p> <p>Tomar las fotos de los exámenes de escritura en la misma posición.</p>

Fuente: Elaboración propia

A continuación, se detallan las actividades y se indican los entregables de cada actividad:

Actividad 1: Identificar bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson.

En esta actividad, se detalla cómo se identifica la primera base de datos. Para el primer *dataset*, se obtuvo la captura de imágenes de escritura de personas, extraídas de la Facultad de Medicina Botucatu, Universidad Estatal de São Paulo, Brasil, en el año 2016. Las cuáles consisten en espirales y meandros escritos por personas tanto sanas como aquellas que padecen la enfermedad de Parkinson. En la Figura 65 se pueden observar ejemplos de cómo es la data.

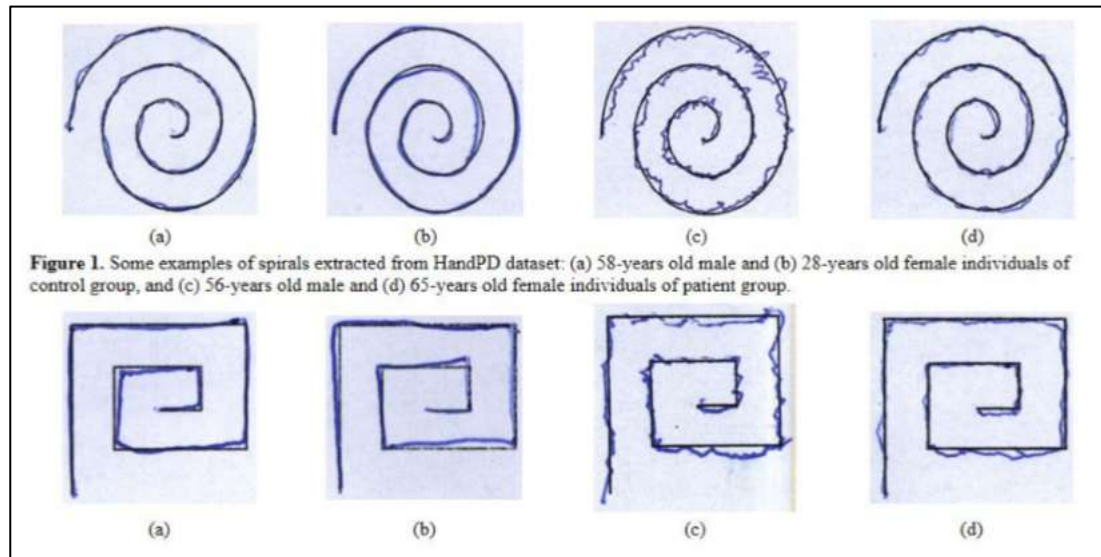


Figura 65. Ejemplos de espirales y meandros de la prueba de Brasil

Fuente: Clayton et al., (2016). *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics*. (p. 3)

Entregable:

- Primer *dataset*, creado en Brasil, que contiene imágenes de escritura de personas (meandros y espirales) que presentan la EP y de personas sanas.

Actividad 2: Crear una base de dato de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson.

En esta actividad, se detalla cómo se crear la segunda base de datos. Para el segundo *dataset*, se hizo una recolección de datos utilizando un examen de escritura (Figura 60). Se hizo un contacto con la Asociación “Actívate Arequipa Liga contra el Parkinson”, donde se tuvo acceso a los nombres y números telefónicos de los pacientes que presentan la Enfermedad de Parkinson, con el debido consentimiento de los mencionados pacientes. De ese modo, se les aplicó el examen para obtener muestras de su escritura, por el contrario, para el caso de recolección de muestras de personas sanas se utilizó la misma cantidad de personas y se les aplicó el mismo examen. En total se recolectaron 1080 imágenes etiquetadas entre personas que

presentan la EP y personas que no, para meandros y espirales respectivamente. Para poder recolectar estos datos se hizo contacto con la presidenta de la Asociación Christine Jeyachandran (ver Anexo 4), en la Figura 66 se puede observar un ejemplo de la muestra.

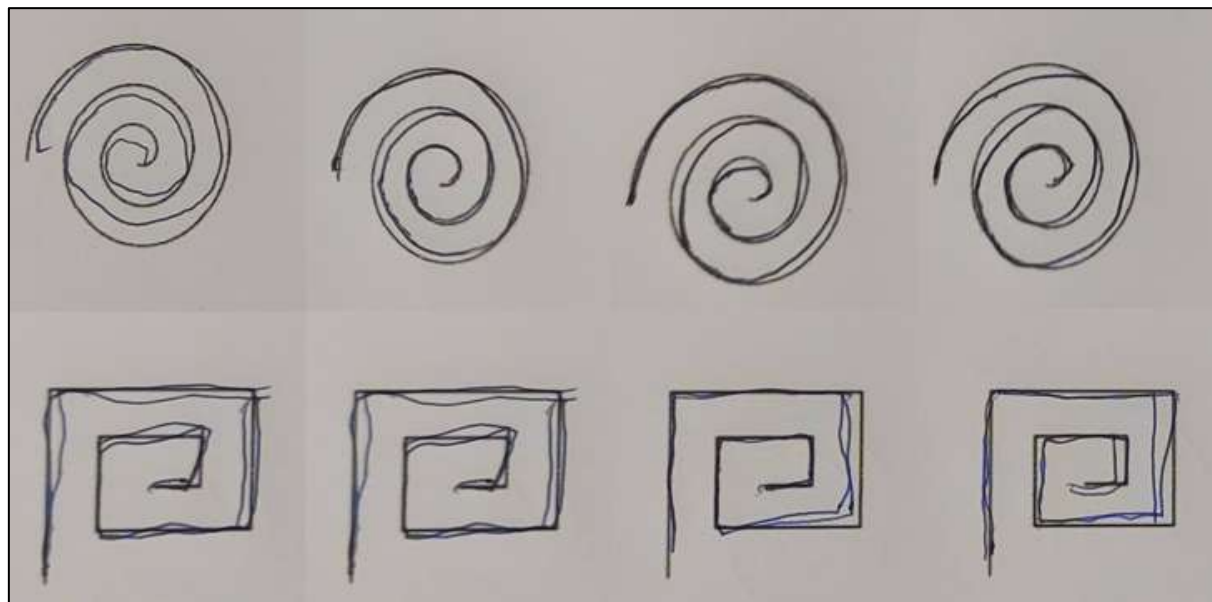


Figura 66. Ejemplos de espirales y meandros de la prueba de Brasil

Fuente: Elaboración propia

Entregable:

- Segundo *dataset*, creado en Perú, que contiene imágenes de escritura de personas (meandros y espirales) que presentan la EP y de personas sanas.

3.5.1.2 preprocesamiento:

En esta etapa se realizará dos preprocesamientos distintos para cada *dataset*, ya que el creado en Brasil a diferencia del creado en Perú, tiene mayor calidad en la imagen, las imágenes ya se encuentran estandarizadas y cada trazo es una imagen. A continuación, se presentarán dos tablas con las tareas y actividades donde se detallará las dos etapas de preprocesamiento:

- **Para el primer *dataset* creado en Brasil:**

Tabla 11.

Actividades etapa de Preprocesamiento primer dataset

Actividades	Descripción	Tareas
1.- Crear nuevas imágenes a partir de las ya existentes	Elaboración de más imágenes para que	Cargar las imágenes.

	posteriormente los modelos de <i>Deep Learning</i> , puedan ser entrenados con una mayor muestra y así tener un aprendizaje más profundo.	Aplicar técnicas de <i>Data Augmentation</i> .
2.- Redimensionar todas las imágenes en una misma escala	Escalar todas las imágenes en una misma dimensión para que puedan ser utilizadas en los modelos de clasificación.	Calcular el promedio de dimensión de todas las imágenes. Redimensionar las imágenes.
3.- Aplicar escala de grises a las imágenes	Pasar todas las imágenes que están en colores a escala de grises.	Aplicar conversión de imágenes en colores a escala de grises.

Fuente: Elaboración propia

A continuación, se detallan las actividades y se indican los entregables de cada actividad:

Actividad 1: Creación de nuevas imágenes a partir de las ya existentes.

En esta actividad, se detalla cómo se crean las nuevas imágenes para aumentar el tamaño del dataset original. Una vez cargadas las imágenes, se hace uso de la técnica *Data Augmentation*, se utilizan conversiones como *flip*, *rotation*, *bright*, *dark*, filtros *min*, *median*.

En detalle, las conversiones realizadas son producto de:

- **Flip y rotation:** Sirve para voltear una matriz 2D alrededor de los ejes vertical, horizontal o ambos.

$$dst_{ij} = \begin{cases} src_{src.rows-i-1,j}, & \text{if } flipCode = 0 \\ src_{i,src.cols-j-1}, & \text{if } flipCode > 0 \\ src_{src.rows-i-1,src.cols-j-1}, & \text{if } flipCode < 0 \end{cases} \quad (44)$$

Donde:

src : imagen de entrada.

dst : matriz de salida del mismo tamaño y tipo que *src*.

flipCode : una bandera para especificar cómo voltear la matriz; 0 significa voltear alrededor del eje xy valor positivo (por ejemplo, 1) significa voltear alrededor del eje y. El valor negativo (por ejemplo, -1) significa invertir ambos ejes.

- **Bright y Dark:** Se hace uso del valor de intensidad mínimo y máximo en una imagen. La idea básica es llevar el valor mínimo (min) a 0 y el máximo (max) a 255.

$$C(x,y) = \frac{(I(x,y) - \min)}{(\max - \min)} * 255 \quad (45)$$

Donde:

$I(x,y)$: Imagen a procesar

$C(x,y)$: Imagen con aumento o disminución lineal de intensidad

- **Filtro min y median:** Se substituye todos los valores de cada pixel central por el mínimo o la mediana de los mencionados valores según sea el caso.

$$\begin{bmatrix} 20 & 20 & 1 & 20 & 10 & 20 \\ 20 & 0 & 0 & 0 & 0 & 0 \\ 20 & 0 & 90 & 90 & 90 & 90 \\ 20 & 0 & 90 & 0 & 90 & 90 \\ 10 & 0 & 90 & 90 & 90 & 90 \\ 10 & 0 & 90 & 90 & 90 & 90 \end{bmatrix} \rightarrow \begin{matrix} \min [0 \ 0 \ 0 \ 1 \ 20 \ 20 \ 20 \ 20 \ 20 \ 90] = 0 \\ \text{median} [0 \ 0 \ 0 \ 1 \ 20 \ 20 \ 20 \ 20 \ 20 \ 90] = 20 \end{matrix}$$

Entregable:

- *Dataset* original multiplicado por la cantidad de nuevas imágenes, que se han generado por imagen

Actividad 2: Redimensionar todas las imágenes en una misma escala

En esta actividad, se detalla cómo se redimensionan las imágenes en la misma escala. Tomando en cuenta que se tienen todas las imágenes cargadas, se calcula el promedio del tamaño de todas las imágenes y posteriormente son redimensionadas, para que todas las imágenes sean del mismo tamaño.

El proceso matemático es:

1. Calcule la relación entre el ancho y el alto
2. Establecer el ancho para que sea igual al límite
3. Calcule la altura en base a la relación calculada previamente

4. Si la altura excede el límite, repita los pasos (2) y (3) usando la altura en su lugar

Por ejemplo, si se tiene una imagen con ancho = 6 y alto = 9 y su umbral es 15.

- La relación entre ancho y alto es 2: 3, en otras palabras,

$$ancho = \frac{2}{3} \times altura \quad (46)$$

- Si aumentamos el ancho para que sea igual al límite, 15, entonces nuestra altura se convierte en:

$$15 \times \frac{3}{2} = 22.5$$

- Como está por encima del límite de 15, en su lugar establecemos la altura en 15, lo que nos da un ancho de:

$$15 \times \frac{2}{3} = 10$$

- Esto está dentro de nuestro límite, por lo que tenemos una nueva imagen con ancho = 10 y alto = 15, que aún obedece a nuestra proporción 2:3 y está dentro del umbral.

En la Figura 67 se puede observar un ejemplo del procedimiento.

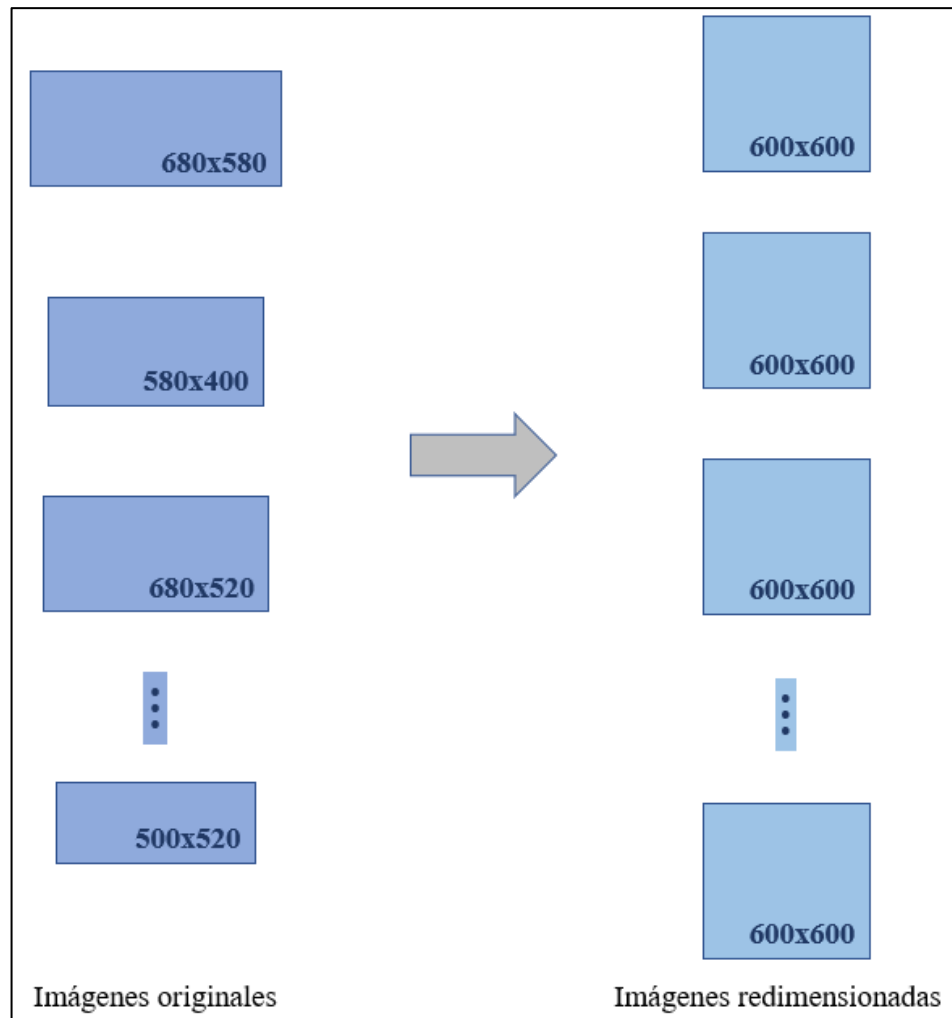


Figura 67. Redimensión de imágenes

Fuente: Elaboración propia

Entregable:

- Dataset con todas las imágenes redimensionadas de un mismo tamaño.

Actividad 3: Aplicar escala de grises a las imágenes

En esta actividad, se detalla cómo se convierte una imagen de colores (RGB) a una imagen en escala de grises.

$$Grayscale = \frac{R+G+B}{3} \quad (47)$$

Donde:

R: Valor promedio de R

G: Valor promedio de G

B: Valor promedio de B

Entregable:

- *Dataset* con todas las imágenes en escala de grises.

- Para el segundo *dataset* creado en Perú:

En el caso de este segundo *dataset* las dos primeras actividades son diferente, sin embargo, las 3 actividades que se mencionaron para el primer *dataset*, son las 3 últimas en este caso. Por lo cual, esas 3 actividades solo se mencionan y no se detallan, ya que el mencionado detalle se puede visualizar en la sección de Actividades 1,2 y 3 del primer *dataset*.

Tabla 12.

Actividades etapa de Preprocesamiento segundo dataset

Actividades	Descripción	Tareas
1.- Cortar las imágenes de los trazos de cada examen de escritura	Creación de un algoritmo que permita cortar los trazos de meandros y espirales, que fueron recolectados a través de los exámenes de escritura aplicados a personas con EP y sanas	Cargar las imágenes de los exámenes de escritura. Crear un algoritmo que corte todos los trazos.
2.- Aplicar técnicas de preprocesamiento para mejorar la calidad de las imágenes	Aplicación de técnicas de preprocesamiento de imágenes como para que a partir de las imágenes se puedan encontrar patrones.	Cargar las nuevas imágenes. Mejorar la calidad de las imágenes a través de técnicas de preprocesamiento.
3.- Crear nuevas imágenes a partir de las ya existentes	Elaboración de más imágenes para que posteriormente los modelos de <i>Deep Learning</i> , puedan ser	Cargar las imágenes. Aplicar técnicas de <i>Data Augmentation</i> .

	entrenados con una mayor muestra y así tener un aprendizaje más profundo.	
4.- Redimensionar todas las imágenes en una misma escala	Escalar todas las imágenes en una misma dimensión para que puedan ser utilizadas en los modelos de clasificación.	Calcular el promedio de dimensión de todas las imágenes. Redimensionar las imágenes.
5.- Aplicar escala de grises a las imágenes	Pasar todas las imágenes que están en colores a escala de grises.	Aplicar conversión de imágenes en colores a escala de grises.

Fuente: Elaboración propia

Actividad 1: Cortar las imágenes de los trazos de cada examen de escritura

En esta actividad, se detalla cómo se cortan las imágenes que fueron tomadas de los exámenes de escritura. Una vez cargadas las imágenes, se construye un algoritmo que corte los trazos, es decir, de un examen de escritura de meandro se obtienen 25 nuevas imágenes, las cuales son los trazos. En la Figura 68 se observa cómo funciona el algoritmo y en la Figura 69 se observa el pseudocódigo del algoritmo.

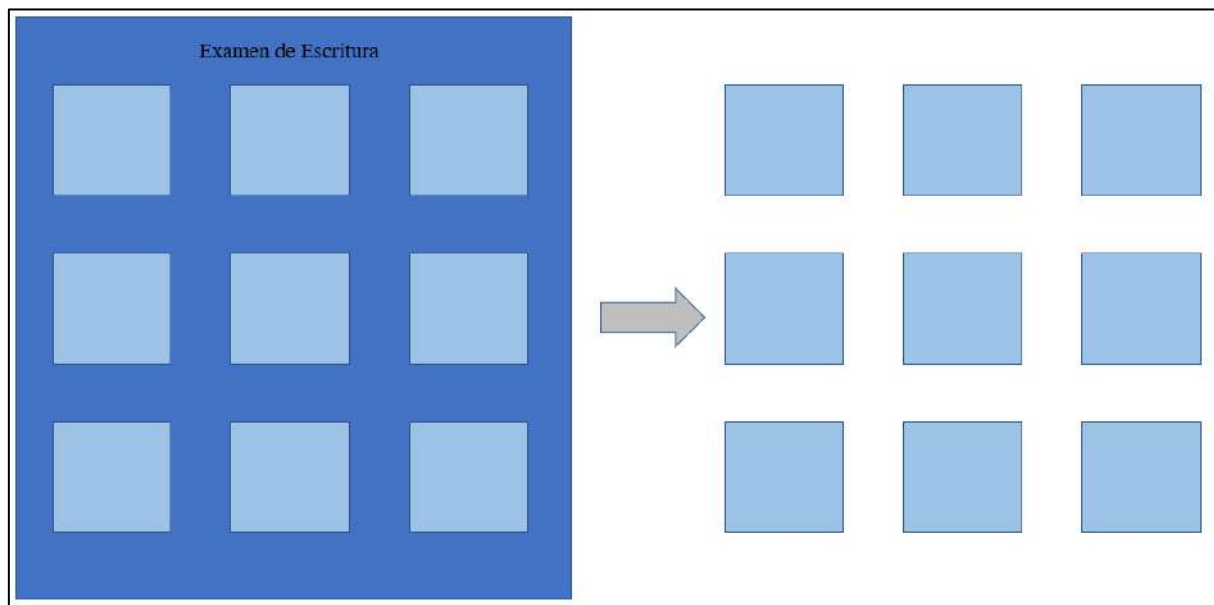


Figura 68. Función del algoritmo que realiza los cortes

Fuente: Elaboración propia

```

1  algoritmo CortarImagen
2
3  variables
4      imagen,coorx1, coory1, ... , coorx20, coory20
5  inicio
6      ^ Repetir
7      | ^ img1 <- imagen[coorx1, coory1]
8      | | img2 <- imagen[coorx2, coory2]
9      | | .
10     | | .
11     | | .
12     | | img19 <- imagen[coorx19, coory19]
13     | | img20 <- imagen[coorx20, coory20]
14     | v
15     v Hasta Que Imagen no existe
16 fin
    
```

Figura 69. Pseudocódigo del algoritmo que realiza los cortes

Fuente: Elaboración propia

Entregable:

- Dataset con las imágenes del examen de escritura recortadas.

Actividad 2: Aplicar técnicas de preprocesamiento para mejorar la calidad de las imágenes

En esta actividad, se detalla cómo se realiza un preprocesamiento a las imágenes para mejorar la calidad de las mismas. Se aplican técnicas como filtro de mediana, contraste y brillo. Las fórmulas de estas técnicas se detallan en la sección de Actividad 1 del primer *dataset*.

Entregable:

- Dataset con mejor calidad de resolución en las imágenes.

Para seguir el proceso se realizan las Actividades 3,4 y 5, el detalle se observa en la sección de Actividades 1,2 y 3 del primer *dataset*. Cuyos **Entregables** son los siguientes, respectivamente para cada Actividad:

Entregable Actividad 3: *Dataset* original multiplicado por la cantidad de nuevas imágenes, que se han generado por imagen

Entregable Actividad 4: *Dataset* con todas las imágenes redimensionadas de un mismo tamaño.

Entregable Actividad 5: *Dataset* con todas las imágenes en escala de grises.

3.5.1.3 extracción de características:

Dado que, las imágenes han sido procesadas y ya se tiene una base de datos completa, están listas para poder usarlas con el fin de obtener vectores característicos, los cuales son las entradas para nuestros modelos de clasificación. En el presente trabajo de investigación se ha decidido usar las técnicas de extracción de características *SURF*, *SIFT*, *ORB*, ya que estas tres técnicas permiten obtener los puntos más característicos de una imagen y los descriptores de la misma. Además, *HOG*, ya que este descriptor alcanza un performance similar a SIFT gracias a las gradientes. A continuación, se presentará una tabla con las tareas y actividades donde se detallará esta etapa de extracción de características:

Tabla 13.

Actividades etapa de extracción de características

Actividades	Descripción	Tareas
1.- Crear el algoritmo <i>Bag of Visual Words</i>	Creación del algoritmo BOVW para transformar los descriptores de una imagen en vectores característicos	Cargar las imágenes. Obtener descriptores. Crear algoritmo BOWV.
2.- Aplicar la técnica de extracción de características SIFT	Obtención de vectores característicos en base a los parámetros del extractor de características SIFT	Parametrizar la función SIFT. Utilizar el algoritmo de Bag of Visual Words. Aplicar la función SIFT.
3.- Aplicar la técnica de extracción de características SURF	Obtención de vectores característicos en base a los parámetros del extractor de características SURF	Parametrizar la función SIFT. Utilizar el algoritmo de Bag of Visual Words. Aplicar la función SURF.

4.- Aplicar la técnica de extracción de características ORB	Obtención de vectores característicos en base a los parámetros del extractor de características ORB	Parametrizar la función SIFT. Utilizar el algoritmo de Bag of Visual Words. Aplicar la función ORB.
5. Aplicar la técnica HOG	Obtención de vectores característicos en base a HOG	Parametrizar la función HOG. Aplicar la función HOG.

Fuente: Elaboración propia

Actividad 1: Crear el algoritmo *Bag of Visual Words*

En esta actividad, se detalla la creación y cómo funciona el algoritmo ***Bag of Visual Words (BOVW)***. Para las tres técnicas mencionadas *SIFT*, *SURF* y *ORB* se obtienen *keypoints* y descriptores respectivamente. Sin embargo, estos descriptores no son los vectores característicos que se necesitan como entrada para los modelos de clasificación. Por lo cual se aplica una técnica conocida como *Bag Of Visual Words (BOVW)*. Esta técnica está basada en *Bag of Words (BOW)* y utiliza los descriptores, para poder representar las imágenes en histogramas según la frecuencia de los descriptores, en la Figura 70 se puede observar cómo es la construcción de los histogramas en base a la frecuencia de los descriptores. (Davida, 2018)

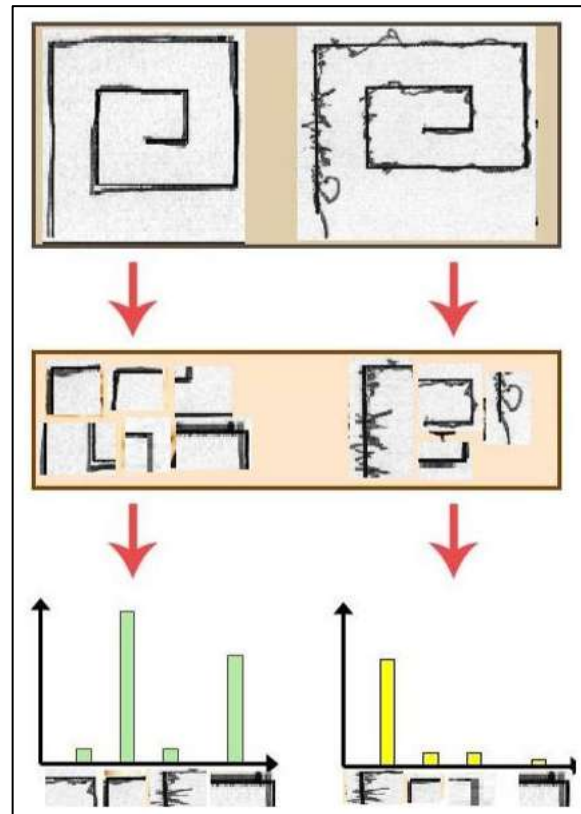


Figura 70. Arquitectura de BOVW

Fuente: Elaboración propia

Seguidamente se aplican algoritmos de agrupamiento como *K-Means* donde cada centroide se utiliza como vocabulario del diccionario visual. En la Figura 71 se puede observar cómo es que los *K-Means* va agrupando a los descriptores.

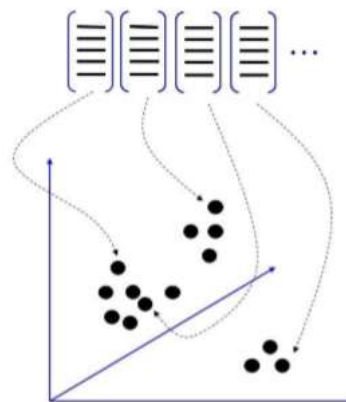


Figura 71. Agrupación de los descriptores con *K-Means*

Fuente: Davida, B. (2018) Agrupación de descriptores

En la Figura 72 se observa el pseudocódigo de BOVW.

```

1  algoritmo BOVW(descriptores, n_clusters)
2
3  variables
4      //descriptores -> salida de extractores de características
5      //                  como SIFT, SURF, ORB
6      //n_clusters -> número de clusters para agrupar
7      descriptores, n_clusters, cont
8  inicio
9      ^   cont <- 0
10     |   agrupacion <- agrupar(descriptores,n_clusters)
11     |   bovw_vector <- []
12     |   Repetir
13     |   ^   cont <- cont + 1
14     |   |   Repetir
15     |   |   ^
16     |   |   |   bovw_vector[index,i] += 1
17     |   |   v
18     |   v   Hasta Que agrupacion[descriptores] no existe
19 v   Hasta Que index, descriptores no existe
20 fin

```

Figura 72. Pseudocódigo del algoritmo BOVW

Fuente: Elaboración propia

Entregable:

- Algoritmo *Bag of Visual Words* (BOVW)

Actividad 2: Aplicar la técnica de extracción de características *SIFT*

En esta actividad, se detalla *SIFT*. *SIFT* es una técnica de extracción de características, la cual da como resultado los puntos más característicos de una imagen. Entre sus principales salidas están los *keypoints*, que vienen a ser las posiciones de los puntos característicos de la imagen y los descriptores que en el caso de *SIFT* son 128. En la Figura 73 se puede observar cuáles son esos puntos más representativos con la técnica *SIFT*. Después estos son el input para el algoritmo BOVW y así se obtienen los vectores característicos.

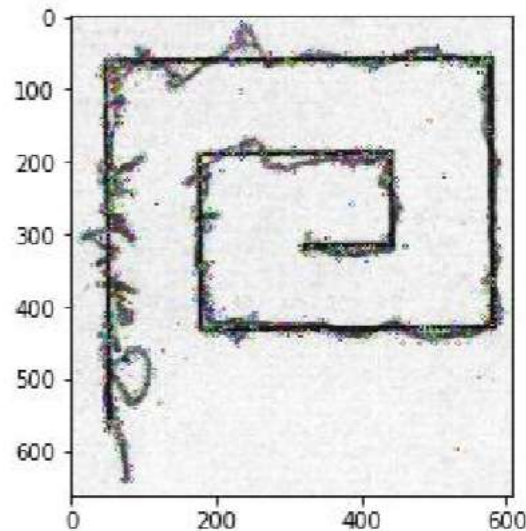


Figura 73. Keypoints en la imagen con *SIFT*

Fuente: Elaboración propia

Entregable:

- Vectores característicos aplicando la técnica *SIFT* a las imágenes.

Actividad 3: Aplicar la técnica de extracción de características *SURF*

En esta actividad, se detalla *SURF*. *SURF* es una técnica de extracción de características, la cual da como resultado los puntos más característicos de una imagen. Entre sus principales salidas están los *keypoints*, que vienen a ser las posiciones de los puntos característicos de la imagen y los descriptores que en el caso de *SURF* son 64. En la Figura 74 se puede observar cuáles son esos puntos más representativos con la técnica *SURF*. Después estos son el input para el algoritmo BOVW y así se obtienen los vectores característicos.

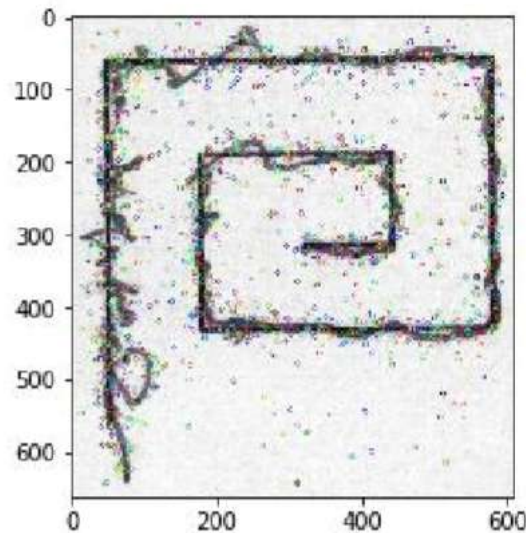


Figura 74. Keypoints en la imagen con *SURF*

Fuente: Elaboración propia

Entregable:

- Vectores característicos aplicando la técnica *SURF* a las imágenes.

Actividad 4: Aplicar la técnica de extracción de características *ORB*

En esta actividad, se detalla *ORB*. *ORB* es una técnica de extracción de características, la cual da como resultado los puntos más característicos de una imagen. Entre sus principales salidas están los *keypoints*, que vienen a ser las posiciones de los puntos característicos de la imagen y los descriptores que en el caso de *ORB* son 32. En la Figura 75 se puede observar cuáles son esos puntos más representativos con la técnica *ORB*. Después estos son el input para el algoritmo BOVW y así se obtienen los vectores característicos.

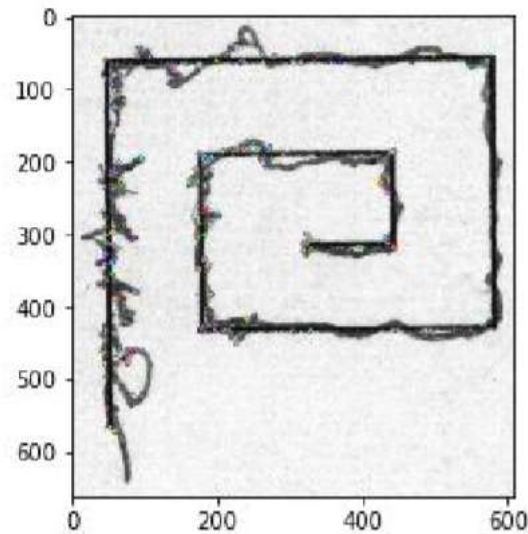


Figura 75. Keypoints en la imagen con ORB

Fuente: Elaboración propia

Entregable:

- Vectores característicos aplicando la técnica *ORB* a las imágenes.

Actividad 4: Aplicar la técnica *HOG*

En esta actividad, se detalla *HOG*. *HOG* es una técnica de extracción de características, la cual da como resultado directamente los vectores característicos que se necesitan para el entrenamiento de los modelos de clasificación. Está basado en los histogramas de gradientes. En la Figura 68 se muestra el diagrama de proceso del funcionamiento de la técnica *HOG*.

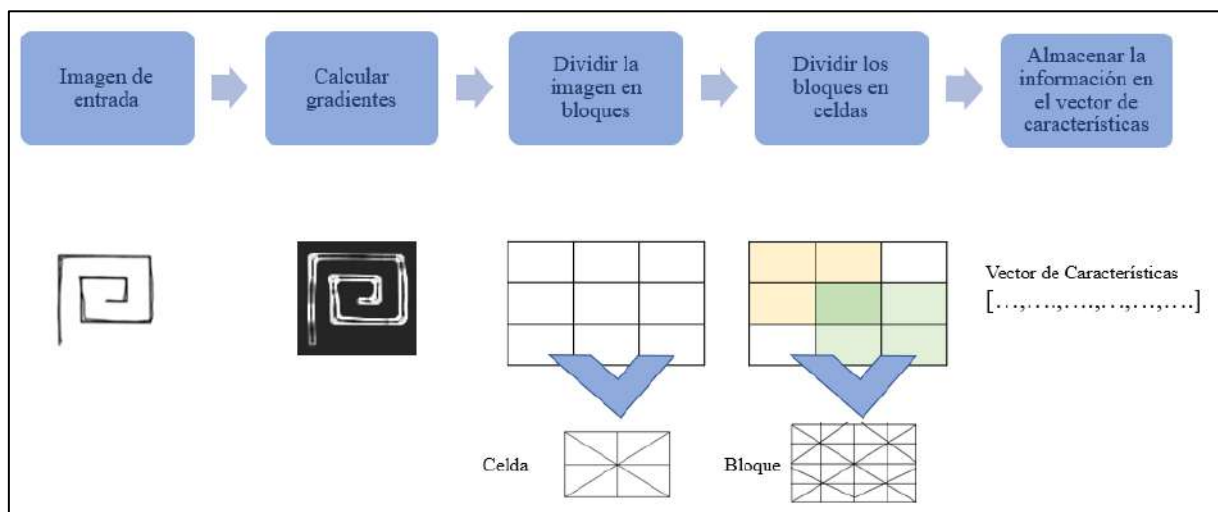


Figura 76. Diagrama de proceso de HOG

Fuente: Elaboración propia

Entregable:

- Vectores característicos aplicando la técnica *HOG* a las imágenes.

3.5.1.4 modelado y clasificación:

En esta etapa se aplican técnicas de machine learning, las cuales son *Support Vector Machine* (SVM), *Random Forest* (RF) y *K-Nearest-Neighbor* (KNN). Además, se utilizan técnicas de *Deep Learning* como redes neuronales convolucionales con diferentes arquitecturas VGG16, VGG19, Inception, ResNet50 y LeNet. A continuación, se presentarán tres tablas con las tareas y actividades donde se detalla la etapa de modelado y clasificación para las Metodologías 1,2 y 3 observadas en la Figura 64.

- **Metodología 1:**

Para esta primera Metodología, la entrada son los vectores característicos que se obtuvieron de la tercera etapa **extracción de vectores característicos**.

Tabla 14.

Actividades etapa de modelado y clasificación – Metodología 1

Actividades	Descripción	Tareas
1.- Asignar <i>target</i> a los vectores característicos	Asignación del <i>target</i> (clase) a los vectores característicos que se obtuvieron de <i>SIFT</i> , <i>SURF</i> , <i>ORB</i> y <i>HOG</i> , según sea el caso 1 si presenta la EP y 0 si es una persona sana	Cargar los vectores característicos en como data estructurada. Asignar 1 o 0 según sea la clase del vector característico.
2.- Desordenar los vectores característicos con su respectiva clase	Aplicación de la función <i>shuffle</i> para que los vectores característicos con su respectiva clase no se encuentren en orden de la mitad 0 y la otra mitad 1	Aplicar función de desordenamiento.

3.- Dividir los vectores característicos (<i>dataset</i>)	División del <i>dataset</i> en <i>train</i> (70%), <i>test</i> (20%) y <i>val</i> (10%) para que este pueda ser entrenado, probado y validado	Crear función para dividir el <i>dataset</i> . Aplicar función para la división del <i>dataset</i> .
4.- Aplicar modelo de <i>Machine Learning</i>	Aplicación del modelo de <i>Machine Learning</i> para la creación del modelo de clasificación	Crear función para aplicar búsqueda de los mejores parámetros. Crear modelo de clasificación con los mejores parámetros para SVM.

Fuente: Elaboración propia

Actividad 1: Asignar *target* a los vectores característicos

En esta actividad, se detalla cómo se realiza la asignación del *target* (clase) a los vectores característicos. Una vez cargados los vectores característicos en Python se tiene identificados que la primera mitad de datos ya sea *SIFT*, *SURF*, *ORB* o *HOG*, pertenecen a la clase “0” es decir, no presentan la EP y la segunda mitad sí presentan la EP. En la Figura 77 se observa el pseudocódigo de la asignación de *target*.

```

1 algoritmo asignación_target
2
3 variables
4     cantidad_imágenes, target
5
6 inicio
7     ^ target <- []
8     | Repetir
9     |     ^ si i < cantidad_imágenes/2
10    |     |     ^
11    |     |     | target[i] <- 0
12    |     |     v
13    |     |     sino
14    |     |     ^
15    |     v     | target[i] <- 1
16    |     |     v
17    v     Hasta Que target < cantidad_imágenes
18 fin

```

Figura 77. Pseudocódigo de la asignación de *target*

Fuente: Elaboración propia

Entregable:

- *Dataframe* con los vectores característicos y el *target* asignado.

Actividad 2: Desordenar los vectores característicos con su respectiva clase

En esta actividad, se detalla cómo se desordenan los vectores característicos con su respectiva clase. Para esto, se utiliza una función de desordenamiento. En la Figura 78 se observa el pseudocódigo de la función de desordenamiento.

```
1 algoritmo desordenamiento
2
3 variables
4     dataset, numero, nuevo_dataset
5
6 inicio
7 ^   Repetir
8 |   ^
9 |   |   nuevo_dataset[i] <= dataset.fila(numero)
10 |   |   numero = numero + 1
11 |   v
12 v   Hasta que numero > dataset.tamaño
13 fin
```

Figura 78. Pseudocódigo de la función de desordenamiento

Fuente: Elaboración propia

Entregable:

- *Dataframe* con los vectores característicos y el *target* desordenados.

Actividad 3: Dividir los vectores característicos (*dataset*)

En esta actividad, se detalla cómo se divide los vectores característicos que se encuentran en un *DataFrame*. En la Figura 79 se observa el pseudocódigo de la función *train_test_val* para poder dividir los datos del mencionado *DataFrame*. Además, se hace uso de la función *train_test_split*.

```

1 algoritmo train_test_val
2
3 variables
4     tabla, cols, X, y, train_ratio, validation_ratio,
5     test_ratio, aux, train, test, val
6
7 algoritmo
8     ^ cols <= tabla.columnas- tabla.TARGET
9     | X <= tabla[cols]
10    | y <= tabla.TARGET
11    |
12    | // Porcentaje de datos por train, test, val
13    | train_ratio <= 0.70
14    | validation_ratio <= 0.10
15    | test_ratio <= 0.20
16    |
17    | // Dividiendo el dataset
18    | aux = tabla.tamaño/3
19    | train = tabla[:aux]
20    | test = tabla[aux:][:aux]
21    | valid = tabla[-aux:]
22 fin
    
```

Figura 79. Pseudocódigo de la función *train_test_val*

Fuente: Elaboración propia

Además, en la Figura 80 se observa de manera gráfica como se divide el *dataset* original en *train*, *test* y *val*.

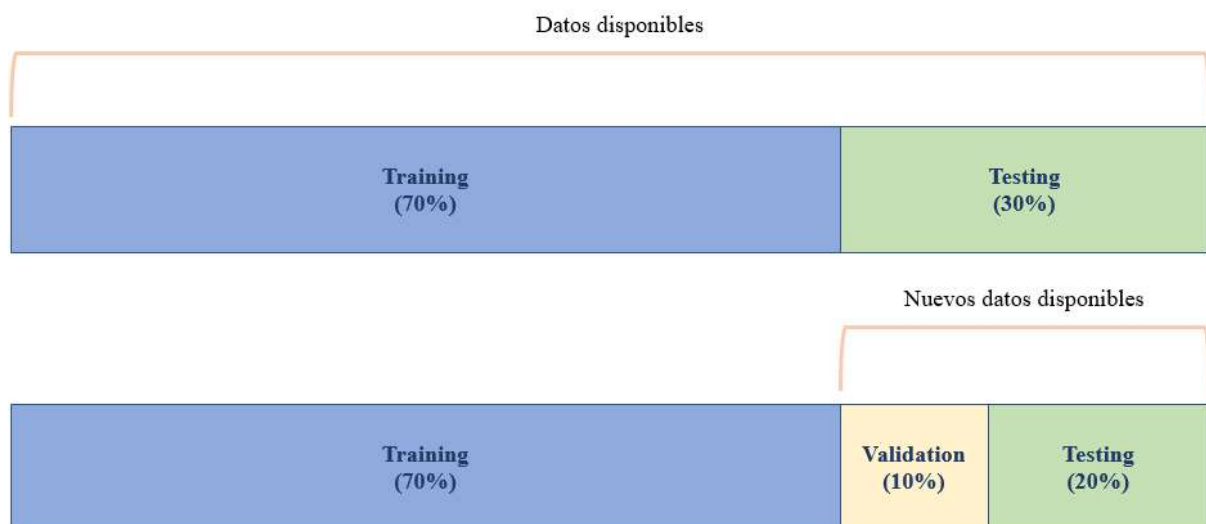


Figura 80. Proceso de la división del dataset en *train*, *test* y *val*

Fuente: Elaboración propia

Entregable:

- Tres *arrays* con los vectores característicos y el *target* asignado divididos en X_{train} , X_{test} , X_{val} , y_{train} , y_{test} , y_{val} .

Actividad 4: Aplicar modelo de Machine Learning

En esta actividad, se detalla cómo se aplica en modelo de *Machine Learning* SVM (Figura 81), además se debe escoger la mejor configuración de parámetros para obtener los mejores resultados y modelo. Por lo cual, se crea la función *GridSearchCV*, cuya función es encontrar los mejores parámetros y evitar el *overfitting*, ya que internamente se utiliza Validación Cruzada. En la Figura 82 se observa el pseudocódigo de la función *GridSearchCV*.

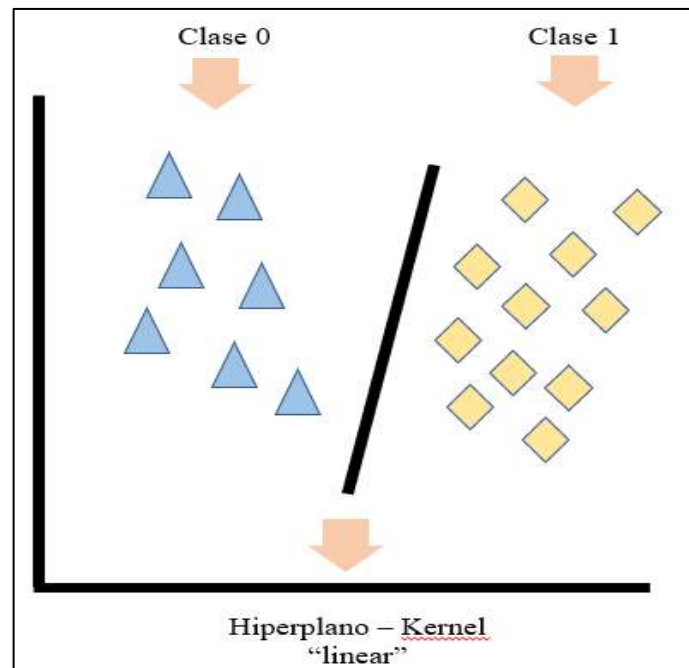


Figura 81. Funcionamiento de SVM

Fuente: Elaboración propia

En la Figura 81 se observa cómo es que un hiperplano lineal se crea para dividir dos clases y así poder discriminar en la clasificación, para entender mejor el funcionamiento de SVM se puede observar la sección 2.2.3.5.2.2 del Marco Teórico.

Rodriguez (2018) indique que:

GridSearchCV es una clase disponible en *scikit-learn* que permite evaluar y seleccionar de forma sistemática los parámetros de un modelo. Indicándole un modelo y los parámetros a probar, puede evaluar el rendimiento del primero en función de los

segundos mediante validación cruzada. El objetivo es escoger el modelo que tenga el mejor rendimiento.

```

1 algoritmo GridSearchCV
2
3 variables
4   X, y, params, model, score, scores
5 inicio
6   ^ params <- {'kernel': ('linear', 'rbf', 'sigmoid'), 'C': [1, 10, 100, 1000], 'gamma': [0.1, 0.01, 0.001, 0.0001]}
7   Repetir
8   |   ^ Repetir
9   |   |   ^ Repetir
10  |   |   |   ^ model = modelo.entrenar(params, X, y)
11  |   |   |   |   score = Total bien clasificados / Total de datos
12  |   |   |   |   v
13  |   |   |   |   Hasta Que params['gamma'] no existe
14  |   |   |   |   scores[i] = score, params
15  |   |   |   v
16  |   |   v   Hasta Que params['C'] no existe
17  |   v   Hasta Que params['kernel'] no existe
18  v   Escribir(max(score), params)
19 fin

```

Figura 82. Pseudocódigo de la función *GridSearchCV*

Fuente: Elaboración propia

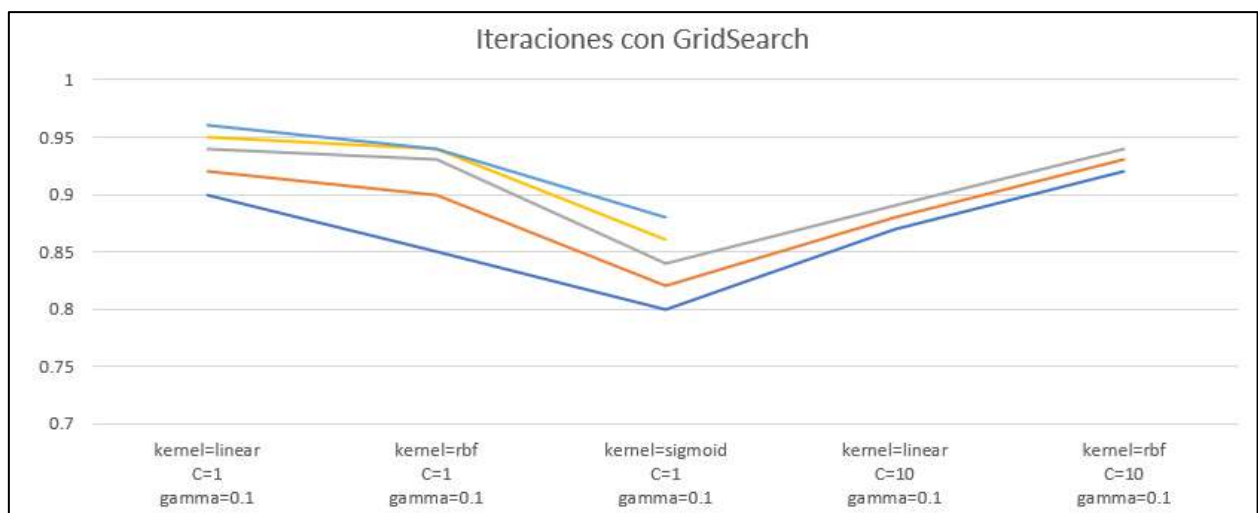


Figura 83. Iteraciones con GridSearch

Fuente: Elaboración propia

Shaikh (2018) explica que para Validación Cruzada:

El procedimiento tiene un solo parámetro llamado k que se refiere al número de grupos en los que se dividirá una muestra de datos determinada. Como tal, el procedimiento a menudo se denomina validación cruzada de k veces. Cuando se elige un valor específico para k , se puede usar en lugar de k en la referencia al modelo, por ejemplo, $k = 10$ se convierte en una validación cruzada de 10 veces.

Si $k = 5$, el conjunto de datos se dividirá en 5 partes iguales y el proceso siguiente se ejecutará 5 veces, cada vez con un conjunto de reserva diferente.

1. Tome el grupo como un conjunto de datos de prueba o reservado
2. Toma los grupos restantes como un conjunto de datos de entrenamiento.
3. Coloque un modelo en el conjunto de entrenamiento y evalúelo en el conjunto de prueba.
4. Conserve la puntuación de la evaluación y descarte el modelo.

Al final del proceso anterior, resuma la habilidad del modelo usando la muestra de puntajes de evaluación del modelo.

En la Figura 84 se observa el funcionamiento de Validación Cruzada para evitar el overffiting.

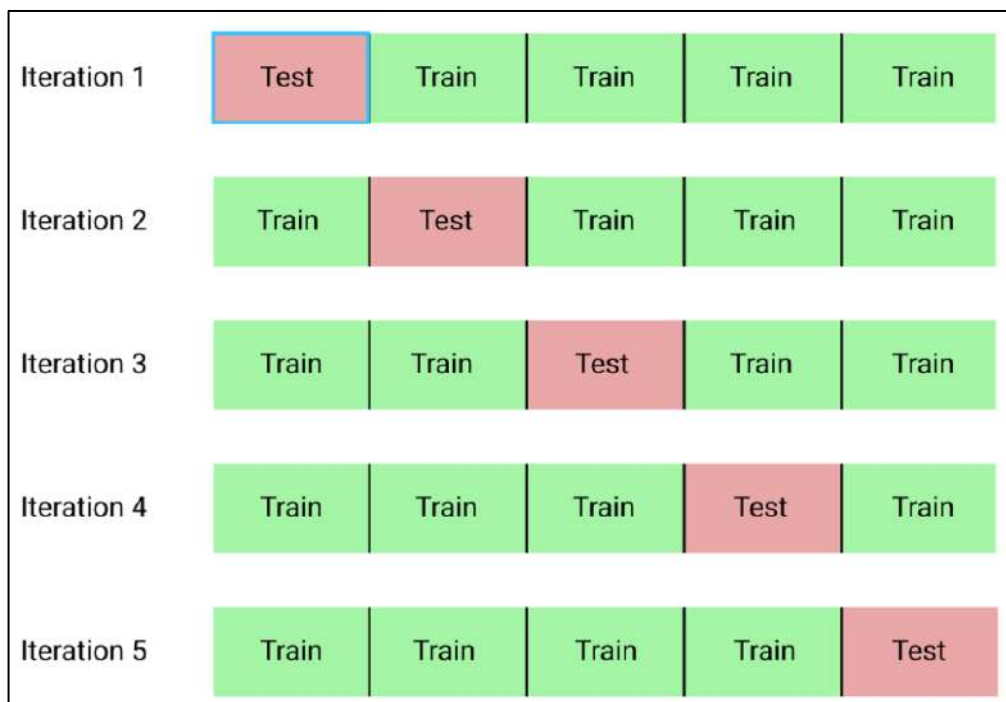


Figura 84. k-Fold Cross Validation

Fuente: Shaikh, R. (2018). *Cross Validation Explained: Evaluating estimator performance*.

Entregable:

- Modelo de clasificación SVM y métricas

- **Metodología 2:**

Para esta segunda Metodología, la entrada son las imágenes pre-procesadas que se obtuvieron de la segunda etapa **preprocesamiento**.

Tabla 15.

Actividades etapa de modelado y clasificación – Metodología 2

Actividades	Descripción	Tareas
1.- Dividir del total de imágenes en train, test y val	División del total de imágenes en <i>train</i> (70%), <i>test</i> (20%) y <i>val</i> (10%) para que este pueda ser entrenado, probado y validado	Cargar las imágenes. Crear función para dividir el total de imágenes. Aplicar función para dividir el total de imágenes.
2.- Aplicar redes neuronales convolucionales de <i>Deep Learning</i>	Aplicación de las redes de <i>Deep Learning</i> para la creación de los modelos de clasificación	Configurar los parámetros de la red convolucional Crear el modelo de clasificación obtenido de la red.

Fuente: Elaboración propia

Actividad 1: Dividir del total de imágenes en train, test y val

En esta actividad, se detalla cómo se divide el total de las imágenes. En la Figura 85 se observa el pseudocódigo de la función *train_test_val_imagenes* para que las mencionadas imágenes puedan ser divididas.

```

1 algoritmo train_test_val_imagenes
2
3 variables
4     total_imagenes, cantidad, train_ratio, validation_ratio, test_ratio
5     train, test, val
6
7 algoritmo
8 ^   cantidad <= tamaño(total_imagenes)
9 |   // Porcentaje de datos por train, test, val
10 |   train_ratio <= 0.70
11 |   validation_ratio <= 0.10
12 |   test_ratio <= 0.20
13 |
14 |   Repetir
15 |   ^   Si i <= cantidad * train_ratio
16 |   |   ^
17 |   |   |   train[i] <= imagen
18 |   |   v
19 |   |   Sino Si i >= cantidad * train_ratio y i <= cantidad * test_ratio
20 |   |   ^
21 |   |   |   test[i] <= imagen
22 |   |   v
23 |   |   Sino Si i >= cantidad * test_ratio y i <= cantidad * validation_ratio
24 |   |   ^
25 |   |   |   val[i] <= imagen
26 |   |   v
27 |   v   i <= i + 1
28 v   Hasta que i <= cantidad
29 fin
    
```

Figura 85. Pseudocódigo de la función *train_test_val_imagenes*

Fuente: Elaboración propia

Entregable:

- Total de imágenes divididas en *train*, *test* y *val*.

Actividad 2: Aplicar redes neuronales convolucionales de Deep Learning

En esta actividad, se detalla cómo se aplican las redes neuronales convolucionales (CNN) con el conjunto de imágenes divididas en *train*, *test* y *val*. Se hace uso de las siguientes CNNs VGG16, VGG19, Inception, Resnet y Lenet. Para las 5 CNNs se debe tener en cuenta los siguientes parámetros:

- **Learning Rate:** Los algoritmos de descenso de gradientes multiplican la gradiente por un escalar conocido como tasa de aprendizaje (o tamaño del paso en algunas ocasiones) para determinar el siguiente punto. Por ejemplo, si la magnitud de la gradiente es 2.5 y la tasa de aprendizaje es 0.01, el algoritmo de descenso de gradientes tomará el siguiente punto 0.025 más alejado del punto anterior.
- **Epochs:** Una Epoch es cuando un conjunto de datos entero se pasa hacia adelante y hacia atrás a través de la red neuronal solo una vez. Dado que una época es demasiado grande para alimentar la computadora a la vez, se divide en varios lotes más pequeños.

- **Batch Size:** Número total de ejemplos de entrenamiento presentes en un solo lote. No puede pasar todo el conjunto de datos a la red neuronal a la vez. Por lo tanto, se divide el conjunto de datos en Número de lotes o conjuntos o partes.

Además, las mencionadas redes tienen arquitecturas diferentes:

- **VGG16**

Para esta red neuronal convolucional se necesita trabajar con las siguientes características:

- Tamaño de imagen de entrada: 224
- Cantidad de convoluciones: 13
- Cantidad de capas pooling: 5
- Cantidad de capas fully connected: 3

Dado que VGG16 fue entrenada para resolver el problema de clasificación de 1000 clases en ImageNet, en sus pesos codifica información para extraer rasgos de muy distintas clases de representadas en las más de 1.4 millones de fotografías de ImageNet. Por lo cual, se hace uso de los pesos entrenados para el presente trabajo de investigación.

Si las capas VGG16 extraen características generales, se puede reusar estas características, las cuales deberán ser pasadas a un nuevo clasificador binario, ya que lo que se busca es predecir si una persona presenta o no la EP.

Para modificar el tamaño del input de la imagen se hace uso de la función *resize*, cuyo proceso se detalla en la Actividad 2 de la etapa de **preprocesamiento**.

En la Figura 86 se explica el proceso de *transfer learning*.

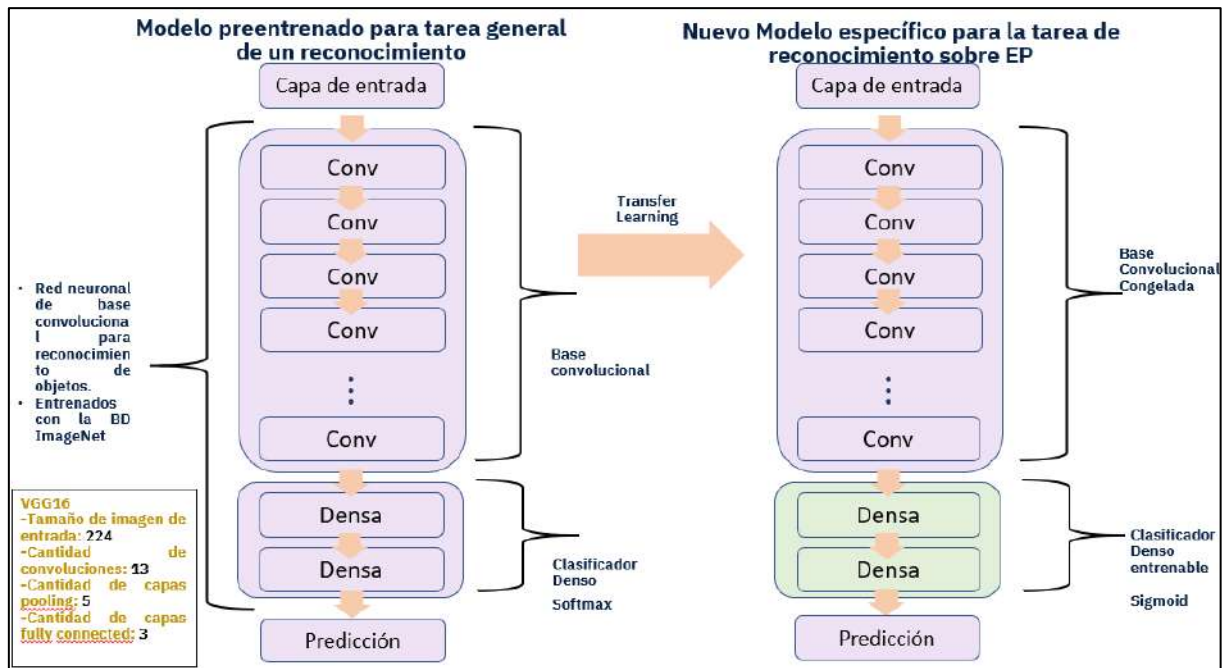


Figura 86. Transfer Learning VGG16

Fuente: Elaboración propia

Como la figura ilustra,

1. Se reusa solo la etapa convolucional de la red.
2. Se define una nueva etapa de clasificación acorde al problema, en este caso se hace uso de una capa sigmoide, ya que la clasificación es binaria.
3. Se fija la subred de extracción de rasgos (se congelan sus pesos) para evitar que sean modificados en el entrenamiento.
4. Se entrenan los pesos “entrenables”, los de la etapa de clasificación haciéndolos pasar los datos por toda la red.

Como se menciona en el punto 2 en la arquitectura original se usa una capa *softmax*, en este caso se usa la función sigmoide.

El valor de la función es cercano a uno de los valores asintóticos, y por eso el valor de la salida está comprendido en la zona baja y alta del sigmoide. Esta función es importante recae en su derivada que siempre es positiva y cercana a cero para los valores grandes positivos o negativos y toma su valor máximo cuando $x=0$. (Ponce, 2010)

$$F_k(u) = \frac{1}{1 + e^{-u}}$$

En la Figura 87 se puede observar la arquitectura de VGG16.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 256, 256, 3)	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0
flatten_1 (Flatten)	(None, 32768)	0
dense_1 (Dense)	(None, 1024)	33555456
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dense_3 (Dense)	(None, 1)	1025

Figura 87. Arquitectura final del modelo de VGG16

Fuente: Elaboración propia

- VGG19:

Para esta red neuronal convolucional se necesita trabajar con las siguientes características:

- Tamaño de imagen de entrada: 224
- Cantidad de convoluciones: 16
- Cantidad de capas pooling: 5
- Cantidad de capas fully connected: 3

Del mismo modo que VGG16, VGG19 fue entrenada para resolver el problema de clasificación de 1000 clases en ImageNet, en sus pesos codifica información para extraer rasgos de muy distintas clases de representadas en las más de 1.4 millones de fotografías de ImageNet. Por lo cual, se hace uso de los pesos entrenados para el presente trabajo de investigación.

Si las capas VGG19 extraen características generales, se puede reusar estas características, las cuales deberán ser pasadas a un nuevo clasificador binario, ya que lo que se busca es predecir si una persona presenta o no la EP.

Para modificar el tamaño del input de la imagen se hace uso de la función *resize*, cuyo proceso se detalla en la Actividad 2 de la etapa de **preprocesamiento**.

En la Figura 88 se explica el proceso de *transfer learning*.

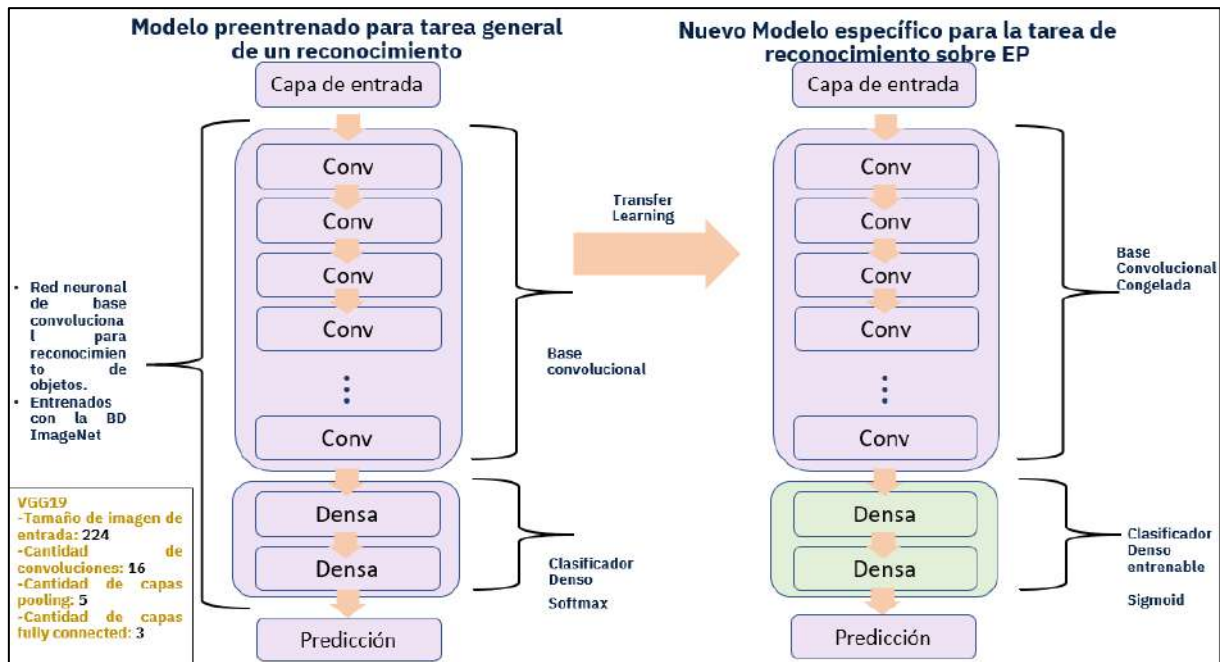


Figura 88. *Transfer Learning* VGG19

Fuente: Elaboración propia

En la Figura 89 se puede observar la arquitectura de VGG19.

Model: "functional_3"		
Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d (Gl	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
dropout_2 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 2)	1026
Total params: 20,550,722		
Trainable params: 20,550,722		
Non-trainable params: 0		

Figura 89. Arquitectura final del modelo de VGG19

Fuente: Elaboración propia

- Inception:

Para esta red neuronal convolucional se necesita trabajar con las siguientes características:

- Tamaño de imagen de entrada: 299
- Cantidad de convoluciones: 98
- Cantidad de capas pooling: 15
- Cantidad de capas fully connected: 2

Del mismo modo que VGG16, Inception fue entrenada para resolver el problema de clasificación de 1000 clases en ImageNet.

Para modificar el tamaño del input de la imagen se hace uso de la función *resize*, cuyo proceso se detalla en la Actividad 2 de la etapa de **preprocesamiento**.

En la Figura 90 se explica el proceso de *transfer learning*.

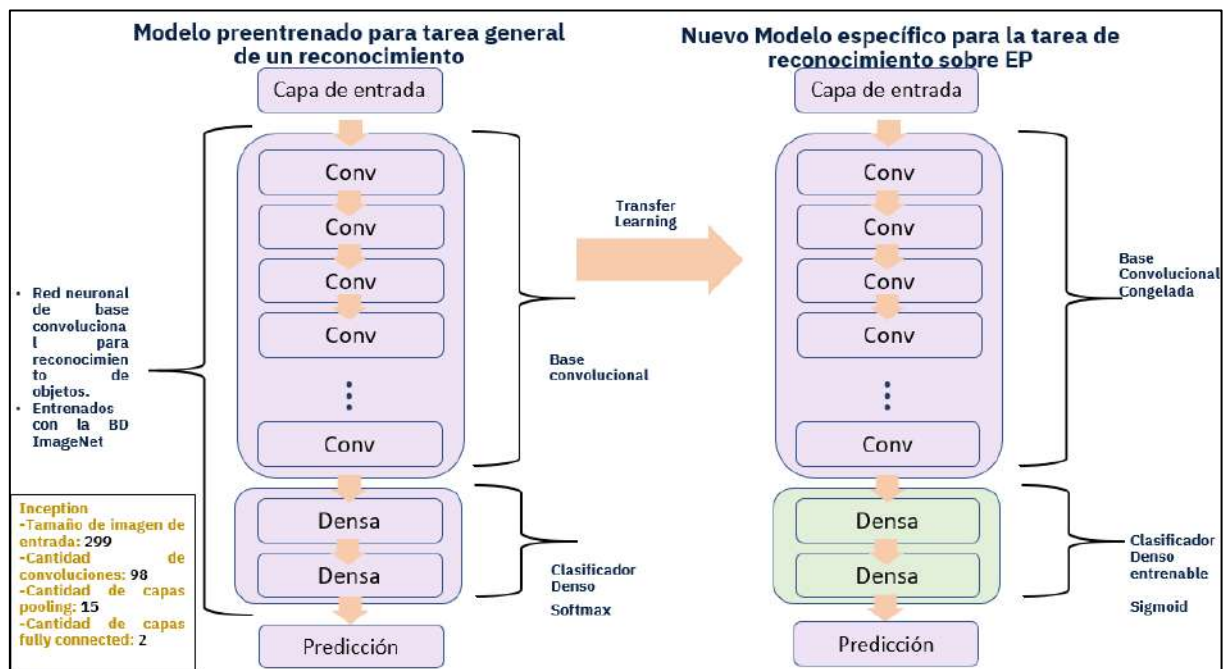


Figura 90. *Transfer Learning* Inception

Fuente: Elaboración propia

En la Figura 91 se puede observar la arquitectura de Inception.

conv2d_92 (Conv2D)	(None, 8, 8, 384)	442368	activation_90[0][0]
average_pooling2d_8 (AveragePool2D)	(None, 8, 8, 2048)	0	mixed9[0][0]
conv2d_85 (Conv2D)	(None, 8, 8, 320)	655360	mixed9[0][0]
batch_normalization_87 (Batch Normalization)	(None, 8, 8, 384)	1152	conv2d_87[0][0]
batch_normalization_88 (Batch Normalization)	(None, 8, 8, 384)	1152	conv2d_88[0][0]
batch_normalization_91 (Batch Normalization)	(None, 8, 8, 384)	1152	conv2d_91[0][0]
batch_normalization_92 (Batch Normalization)	(None, 8, 8, 384)	1152	conv2d_92[0][0]
conv2d_93 (Conv2D)	(None, 8, 8, 192)	393216	average_pooling2d_8[0][0]
batch_normalization_85 (Batch Normalization)	(None, 8, 8, 320)	960	conv2d_85[0][0]
activation_87 (Activation)	(None, 8, 8, 384)	0	batch_normalization_87[0][0]
activation_88 (Activation)	(None, 8, 8, 384)	0	batch_normalization_88[0][0]
activation_91 (Activation)	(None, 8, 8, 384)	0	batch_normalization_91[0][0]
activation_92 (Activation)	(None, 8, 8, 384)	0	batch_normalization_92[0][0]
batch_normalization_93 (Batch Normalization)	(None, 8, 8, 192)	576	conv2d_93[0][0]
activation_85 (Activation)	(None, 8, 8, 320)	0	batch_normalization_85[0][0]
mixed9_1 (Concatenate)	(None, 8, 8, 768)	0	activation_87[0][0] activation_88[0][0]
concatenate_1 (Concatenate)	(None, 8, 8, 768)	0	activation_91[0][0] activation_92[0][0]
activation_93 (Activation)	(None, 8, 8, 192)	0	batch_normalization_93[0][0]
mixed10 (Concatenate)	(None, 8, 8, 2048)	0	activation_85[0][0] mixed9_1[0][0] concatenate_1[0][0] activation_93[0][0]
flatten_1 (Flatten)	(None, 131072)	0	mixed10[0][0]
dense_4 (Dense)	(None, 1024)	134218752	flatten_1[0][0]
dropout_3 (Dropout)	(None, 1024)	0	dense_4[0][0]
dense_5 (Dense)	(None, 1024)	1049600	dropout_3[0][0]
dropout_4 (Dropout)	(None, 1024)	0	dense_5[0][0]
dense_6 (Dense)	(None, 1)	1025	dropout_4[0][0]
Total params: 157,072,161			
Trainable params: 135,269,377			
Non-trainable params: 21,802,784			

Figura 91. Arquitectura final del modelo de Inception

Fuente: Elaboración propia

- Resnet:

Para esta red neuronal convolucional se necesita trabajar con las siguientes características:

- Tamaño de imagen de entrada: 224

- Cantidad de convoluciones: 25
- Cantidad de capas pooling: 10
- Cantidad de capas fully connected: 3

Debido al tamaño de la arquitectura ResNet, por la cantidad de capas internas que posee, ingresar a este link: <https://github.com/virginiamonroy/Tesis-VMM-2020/blob/main/Arquitectura%20ResNet/Arquitectura%20ResNet.pdf>

Del mismo modo que VGG16, ResNet fue entrenada para resolver el problema de clasificación de 1000 clases en ImageNet.

Para modificar el tamaño del input de la imagen se hace uso de la función *resize*, cuyo proceso se detalla en la Actividad 2 de la etapa de **preprocesamiento**.

En la Figura 92 se explica el proceso de *transfer learning*.

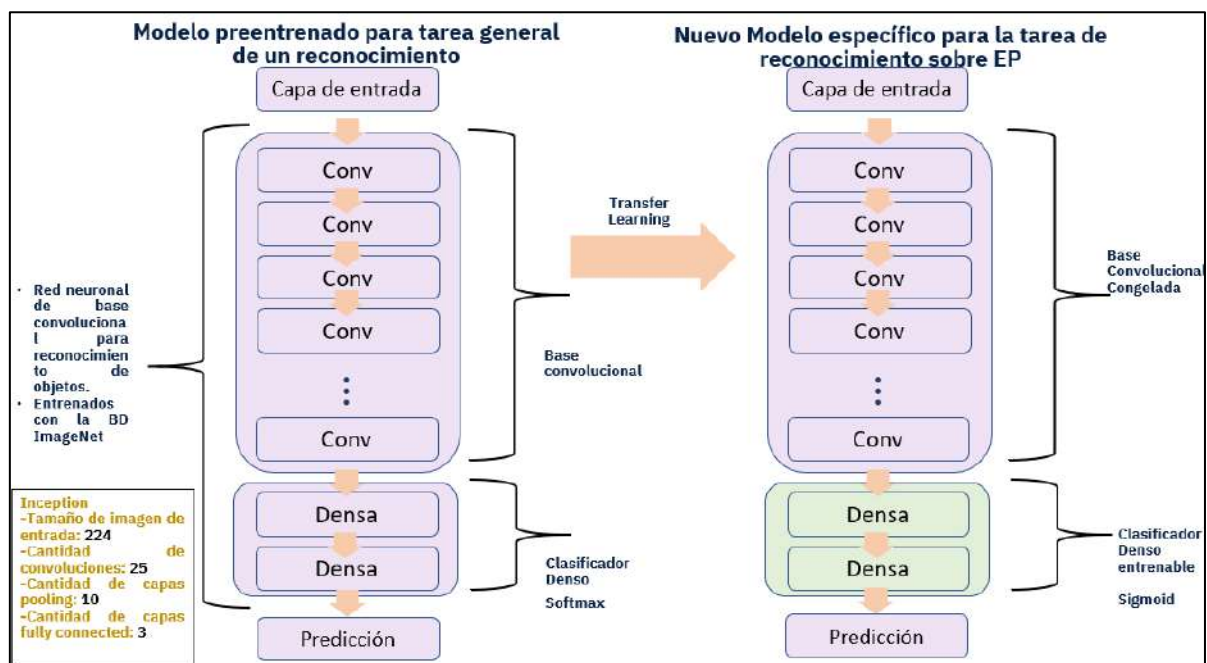


Figura 92. *Transfer Learning* Resnet

Fuente: Elaboración propia

- LeNet:

Para esta red neuronal convolucional se necesita trabajar con las siguientes características:

- Tamaño de imagen de entrada: 28
- Cantidad de convoluciones: 3

- Cantidad de capas pooling: 2
- Cantidad de capas fully connected: 2

En este caso no se hace *transfer learning*.

En la Figura 93 se puede observar la arquitectura de LeNet.

Model: "sequential_8"		
Layer (type)	Output Shape	Param #
conv2d_21 (Conv2D)	(None, 254, 254, 6)	168
average_pooling2d_21 (Average Pooling)	(None, 127, 127, 6)	0
conv2d_22 (Conv2D)	(None, 125, 125, 16)	880
average_pooling2d_22 (Average Pooling)	(None, 62, 62, 16)	0
flatten_8 (Flatten)	(None, 61504)	0
dense_24 (Dense)	(None, 120)	7380600
dense_25 (Dense)	(None, 84)	10164
dense_26 (Dense)	(None, 2)	170
Total params: 7,391,982		
Trainable params: 7,391,982		
Non-trainable params: 0		

Figura 93. Arquitectura final del modelo de LeNet

Fuente: Elaboración propia

En la Figura 94 se puede observar cómo es el proceso de entrenamiento de una CNN. Para entender a detalle cómo se desarrolla el entrenamiento de una CNN observar la sección 2.2.3.2 en el Marco Teórico.

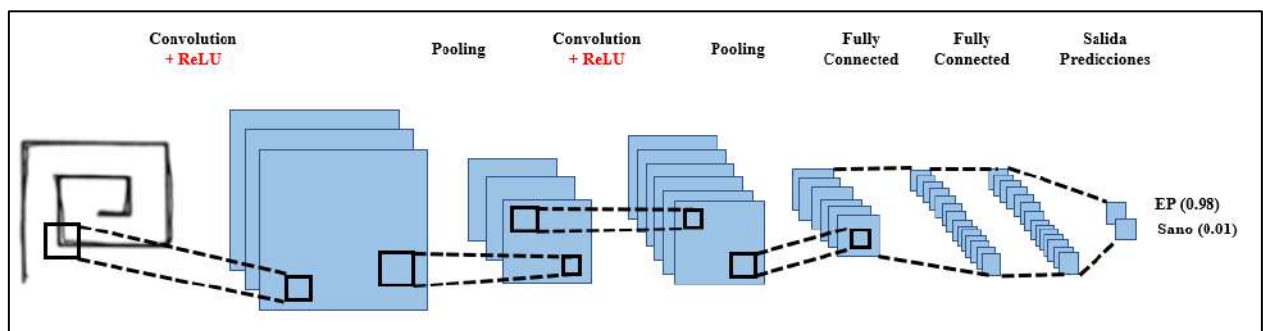


Figura 94. Proceso de entrenamiento de una CNN

Fuente: Elaboración propia

Entregable:

- Modelos de CNNs con sus respectivas métricas

- **Metodología 3:**

Para esta tercera Metodología, las entradas son la penúltima capa de las CNNs que se utilizan en la Metodología 2, las cuales son: VGG16, VGG19, Inception, Resnet y Lenet.

Tabla 16.

Actividades etapa de modelado y clasificación – Metodología 3

Actividades	Descripción	Tareas
1.- Obtener las penúltimas capas de las CNNs	Obtención de la penúltima capa de una CNN para poder extraer los vectores característicos antes de que pasen por la capa de clasificación	Cargar las CNNs generadas en la Metodología 2. Obtener las penúltimas capas de las CNNs.
2.- Asignar <i>target</i> a los vectores característicos	Asignación del <i>target</i> (clase) a los vectores característicos que se obtuvieron de <i>SIFT</i> , <i>SURF</i> , <i>ORB</i> y <i>HOG</i> , según sea el caso 1 si presenta la EP y 0 si es una persona sana	Cargar los vectores característicos en como data estructurada. Asignar 1 o 0 según sea la clase del vector característico.
3.- Desordenar los vectores característicos con su respectiva clase	Aplicación de la función <i>shuffle</i> para que los vectores característicos con su respectiva clase no se encuentren en orden de la mitad 0 y la otra mitad 1	Aplicar la función de desordenamiento.
4.- Dividir los vectores característicos (<i>dataset</i>)	División del <i>dataset</i> en <i>train</i> (70%), <i>test</i> (20%) y <i>val</i> (10%) para que este pueda ser entrenado, probado y validado	Crear función para dividir el <i>dataset</i> . Aplicar función para la división del <i>dataset</i> .

5.- Aplicar modelos de <i>Machine Learning</i>	Aplicación de los modelos de <i>Machine Learning</i> para la creación de los modelos de clasificación	<p>Crear funciones para aplicar <i>GridSearchCV</i>.</p> <p>Crear modelo de clasificación con los mejores parámetros para SVM, RF y KNN.</p>
--	---	--

Fuente: Elaboración propia

Actividad 1: Obtener la penúltima capa de la CNN

En esta actividad, se detalla cómo se extrae la penúltima capa de una CNN. En la Figura 91 se observa el proceso de extracción. En la Figura se observa que el proceso de las CNN es el mismo, la diferencia es que antes de entrar a la capa de clasificación, esta es obtenida como valores que representan los vectores característicos, para luego ser cargados en una tabla como data estructura.

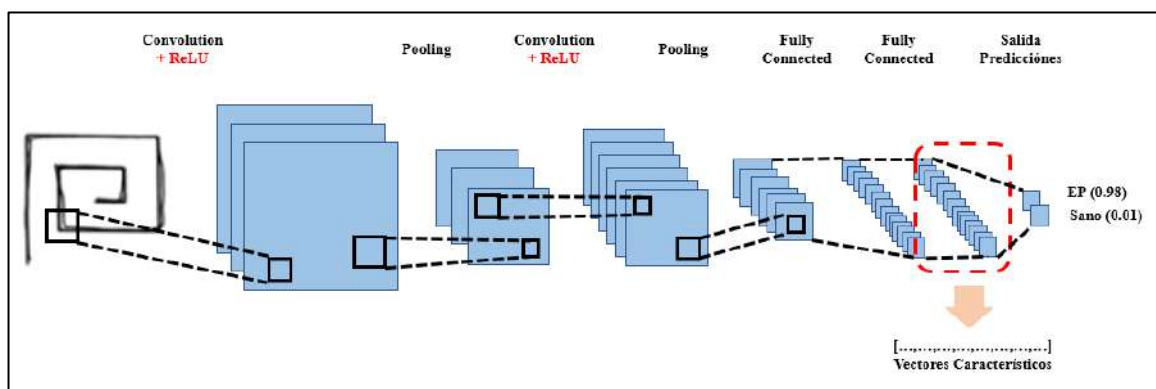


Figura 95. Proceso de entrenamiento de una CNN

Fuente: Elaboración propia

Entregable:

- Vectores Característicos de las CNNs

Para seguir el proceso se realizan las Actividades 2, 3, y 4, el detalle se observa en la sección de Actividades 1, 2, y 3 de la primera Metodología. Cuyos **Entregables** son los siguientes, respectivamente para cada Actividad:

Entregable Actividad 2: *Dataframe* con los vectores característicos y el *target* asignado.

Entregable Actividad 3: *Dataframe* con los vectores característicos y el *target* desordenados.

Entregable Actividad 4: Tres arrays con los vectores característicos y el *target* asignado divididos en *X_train*, *X_test*, *X_val*, *y_train*, *y_test*, *y_val*.

Actividad 5: Aplicar modelos de *Machine Learning*

En esta actividad, se detalla cómo se aplican los modelos de *Machine learning* RF y KNN. Es importante resaltar, que el modelo SVM se detalla en la Actividad 4 de la Metodología 1.

- KNN:

Harrison (2018) indica que:

El algoritmo KNN asume que existen cosas similares en las proximidades. En otras palabras, cosas similares están cerca unas de otras.

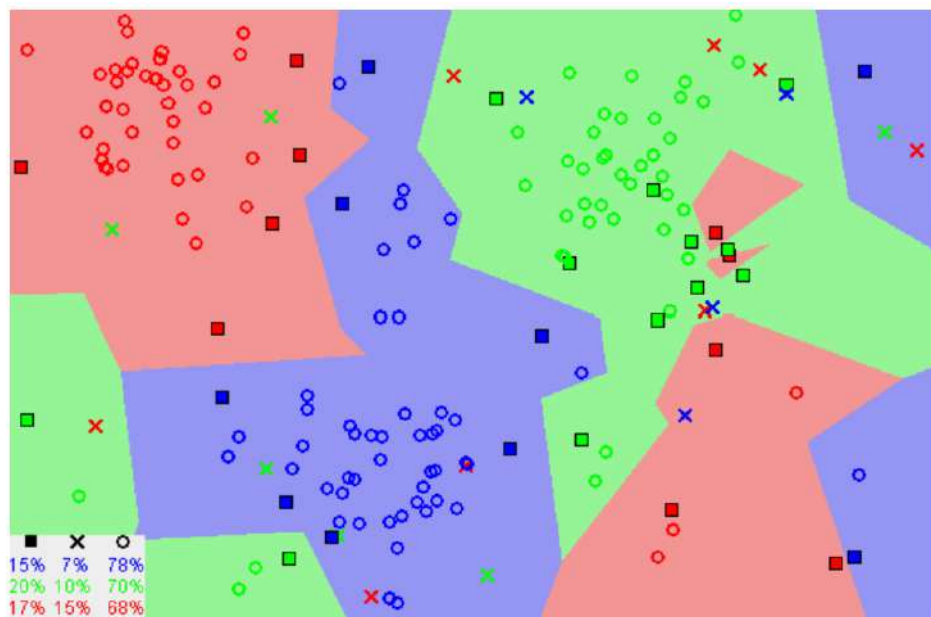


Figura 96. Ejemplo de aplicación de KNN

Fuente: Harrison, O. (2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*

1. Cargar los datos
2. Inicializar K a su número elegido de vecinos
3. Para cada ejemplo en los datos

- 3.1. Calcule la distancia entre el ejemplo de consulta y el ejemplo actual a partir de los datos.
- 3.2. Agregar la distancia y el índice del ejemplo a una colección ordenada
4. Ordene la colección ordenada de distancias e índices de menor a mayor (en orden ascendente) por las distancias
5. Elija las primeras K entradas de la colección ordenada
6. Obtenga las etiquetas de las entradas K seleccionadas
7. Si hay regresión, devuelva la media de las etiquetas K
8. Si es clasificación, devuelva la moda de las etiquetas K

- **RF:**

Random Forest, como su nombre lo indica, consta de una gran cantidad de árboles de decisión individuales que operan como un conjunto. Cada árbol individual en el bosque aleatorio devuelve una predicción de clase y la clase con más votos se convierte en la predicción del modelo.

1. Crear de un conjunto de datos bootstrap para cada árbol

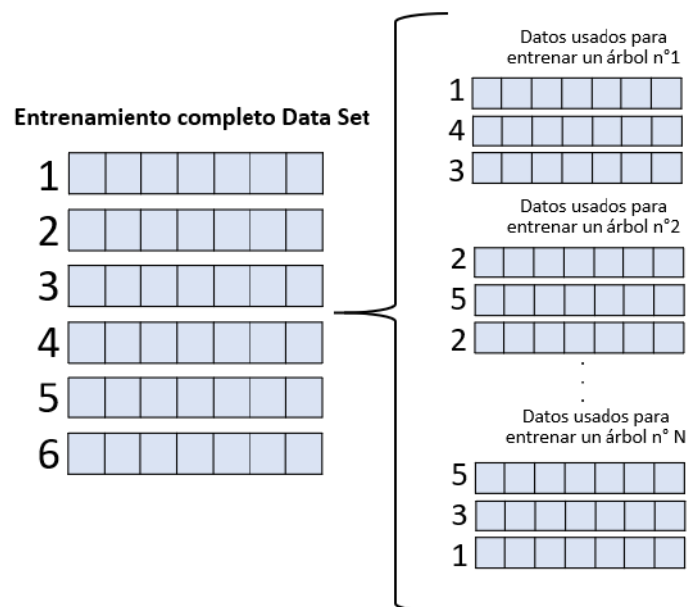


Figura 97. Creación de conjuntos de datos individuales.

Fuente: Elaboración propia

2. Entrenar un bosque de árboles usando estos conjuntos de datos aleatorios y agregue un poco más de aleatoriedad con la selección de características

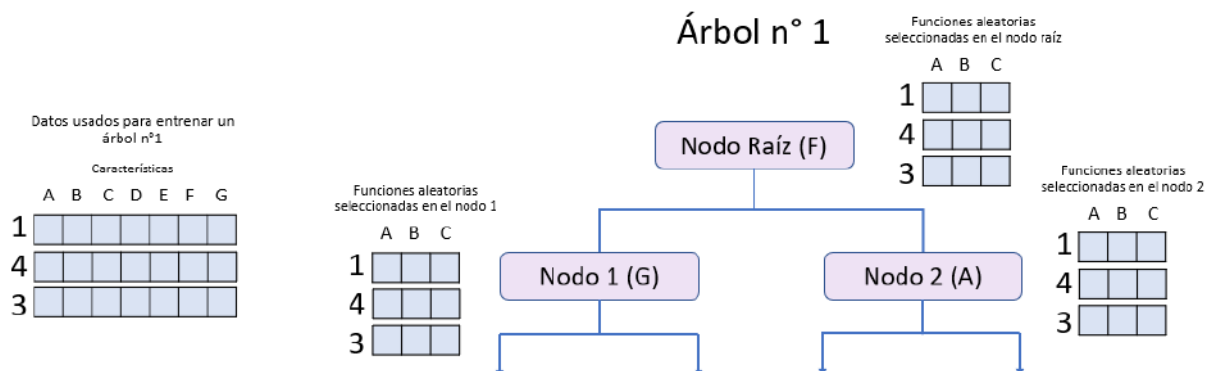


Figura 98. Selección aleatoria de características para el árbol de construcción

Fuente: Elaboración propia

3. Repetir el proceso para los árboles N
4. Hacer predicciones usando un bosque aleatorio (Zornoza, 2020)

Problema de clasificación: pre-diagnóstico médico



Figura 99. Predicciones usando un bosque aleatorio

Fuente: Elaboración propia

Entregable:

- Modelos de clasificación SVM, RF, KNN y métricas.

3.5.2 metodología para la medición de resultados

El análisis de los resultados de la implementación se realizó mediante la evaluación de la relación entre las variables definidas buscando medir la efectividad de modelo de clasificación.

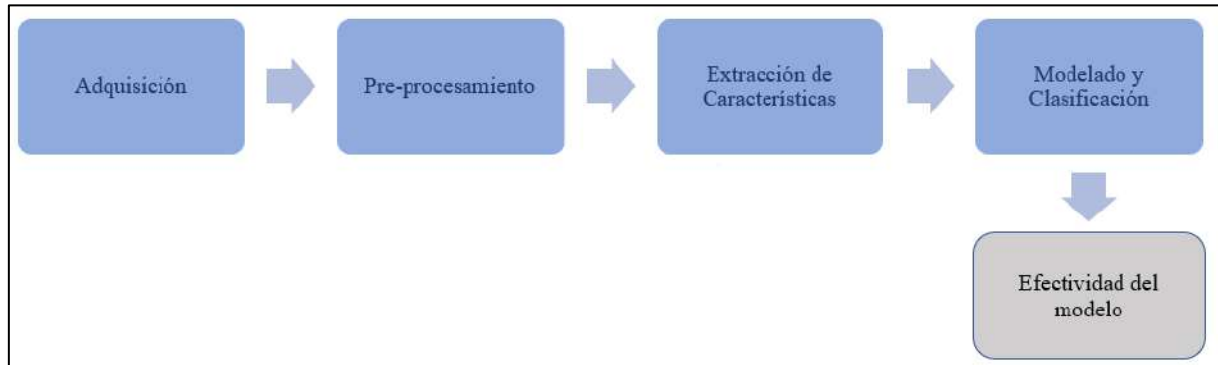


Figura 100. Medición de resultados de implementación

Fuente: Elaboración propia

Una vez entrenados los modelos de *Machine Learning*, proceden a ser evaluados a través de las siguientes métricas:

- Accuracy:

El *Accuracy* es una métrica que muestra la relación entre las observaciones predichas correctamente sobre el total de observaciones. Es importante resaltar, que esta métrica es útil cuando el conjunto de datos (falsos positivos y falsos negativos) son simétricos, ya que de no ser el caso el *Accuracy* ya no es una métrica del todo confiable. (Joshi, 2016)

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (48)$$

- Precision:

La *Precision* es una métrica que muestra la relación entre las observaciones que han sido predichas correctamente sobre el total de observaciones positivas predichas. El tener una alta precisión significa que existe una baja tasa de falsos positivos. (Joshi, 2016)

$$Precision = \frac{TP}{TP+FP} \quad (49)$$

- Recall:

El *Recall* es una métrica que muestra la relación entre las observaciones que han sido predichas correctamente sobre el total de observaciones de la clase real “*positive*”. (Joshi, 2016)

$$Recall = \frac{TP}{TP+FN} \quad (50)$$

- F1:

F1 es el promedio ponderado de Precision y Recall. Por lo cual, esta métrica toma en cuenta a los falsos positivos y a los falsos negativos. Cuando se tiene un conjunto de observaciones que no presentan simetría, es decir, dichas observaciones están desbalanceadas, se recomienda usar la métrica F1. (Joshi, 2016)

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (51)$$

Donde:

TP = True Positives: Son aquellos valores positivos predichos correctamente.

FP = False Positives: Son aquellos valores negativos predichos correctamente.

TN = True Negatives: Son aquellos valores positivos predichos incorrectamente.

FN = False Negatives: Son aquellos valores negativos predichos incorrectamente.

- ROC AUC:

Narkhede (2018) sostiene que:

La curva AUC - ROC es una medida de rendimiento para los problemas de clasificación en varios valores de umbral. ROC es una curva de probabilidad y AUC representa el grado o medida de separabilidad. Indica cuánto es capaz el modelo de distinguir entre clases. Cuanto mayor sea el AUC, mejor será el modelo para predecir 0 como 0 y 1 como 1. Por analogía, cuanto mayor sea el AUC, mejor será el modelo para distinguir entre dos clases.

Un modelo excelente tiene un AUC cercano al 1, lo que significa que tiene una buena medida de separabilidad. Un modelo deficiente tiene un AUC cercano al 0, lo que significa que tiene la peor medida de separabilidad. De hecho, significa que está correspondiendo el resultado. Predice 0 como 1 y 1 como 0. Y cuando el AUC es 0,5, significa que el modelo no tiene capacidad de separación de clases en absoluto.

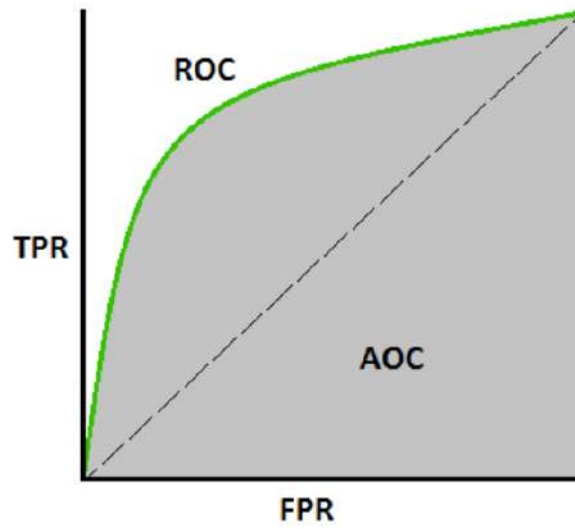


Figura 101. Curva ROC

Fuente: Narkhede, S. (2018). *Understanding AUC - ROC Curve*

$$TPR = \frac{TP}{TP+FN} \quad (52)$$

$$FPR = \frac{FP}{TN+FP} \quad (53)$$

3.6 Cronograma de Actividades

Tabla 17.

Cronograma de Actividades

ACTIVIDADES /TAREAS	Abril				Mayo				Junio				Julio				Agosto				Setiembre				Octubre				Noviembre				Diciembre			
	Mes1				Mes2				Mes2				Mes4				Mes5				Mes6				Mes7				Mes8				Mes9			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
INICIO																																				
Selección de tema de Investigación																																				
Redacción del título																																				
Elaboración del Plan de Investigación																																				
Identificación de Realidad Problemática																																				
Planteamiento del Problema																																				
Planteamiento de los Objetivos																																				
Justificación y Delimitación																																				
DESARROLLO																																				
Revisión Bibliográfica																																				
Marco Teórico																																				
Recolección de Datos																																				
Planteamiento de Hipótesis																																				
Metodología																																				
Recolección de los Datos																																				
Preparación de los Datos																																				
Análisis de Datos																																				
Desarrollo de la Solución Analítica																																				
Evaluación Económica y Financiera																																				
Discusión sobre los Resultados																																				
Despliegue																																				
Conclusiones y Recomendaciones																																				
CIERRE																																				
Elaboración del Informe Final																																				
Presentación																																				
Sustentación																																				

Fuente: Elaboración propia

Nota: Se destinó entre dos a tres horas diarias de investigación para la realización de la tesis

Actividades Cumplidas



Actividades por cumplir



3.7 Presupuesto

En este punto se determinó una estimación de los recursos que se invirtió para el desarrollo del plan de tesis.

Tabla 18.

Presupuesto

Tipo	Categoría	Recurso	Descripción	Fuente Financiadora	Monto
Recursos Disponibles	Equipo	Laptop	Procesar información, mantenimiento	Propio	S/. 40
	Software	Python	Para procesar las imágenes y construir el modelo	Propio	-
Recursos no Disponibles	Consultoría	Humano	Para poder obtener conocimiento de expertos en el tema	Propio	S/. 150
	Movilidad	Pasajes	Para poder movilizarme	Propio	S/. 50
	Útiles de Escritorio	Hojas e impresiones	Para la elaboración de los exámenes de escritura	Propio	S/. 40
Recursos de Desarrollo e Implementación	Cloud	Cloud Foundry	Para el despliegue de la prueba en línea y la página web	Propio	S/.100
	Cloud	Watson Studio	Para entrenar las redes con GPUs	Propio	S/. 40
Total					S/. 420

Fuente: Elaboración propia

Adicionalmente, existen ciertos costos de algunos servicios cuyos planes *Lite* soportan el consumo actual que se utilizó en el presente trabajo de investigación.

Capítulo 4: Desarrollo del Experimento

En el presente capítulo se detalla cómo es que se realizó todo el desarrollo del experimento, lo cual involucra la adquisición del *dataset*, el preprocesamiento de la data, la extracción de características y los modelos que se utilizaron para la predicción.

4.1 Adquisición del Dataset

En esta etapa se describe a detalle la adquisición de los dos *dataset* con los cuales se trabajó en el presente trabajo de investigación:

Para el primer *dataset*, se tiene la captura de imágenes de escritura de personas, en el año 2016 extraídas de la Facultad de Medicina Botucatu, Universidad Estatal de São Paulo, Brasil, las cuáles consisten en espirales y meandros dibujados por personas tanto sanas como aquellas que padecen la enfermedad de Parkinson. Se utilizaron dos *datasets* que provienen de la misma fuente, llamados *HandPD* y *NewHandPD*. En la Tabla 19 se puede observar cuantos datos se recolectaron en total tanto de personas sanas como de personas que presentan la EP.

Tabla 19.

Estructura de base de datos

	HAND PD			NEW HAND PD			TOTAL	
EP			EP			EP		
	MEANDER	296		MEANDER	124		MEANDER	420
	ESPIRALES	296		ESPIRALES	124		ESPIRALES	420
SANOS			SANOS			SANOS		
	MEANDER	72		MEANDER	140		MEANDER	212
	ESPIRALES	72		ESPIRALES	140		ESPIRALES	212

Fuente: Elaboración propia

Se puede observar en la Tabla 19 que al unir la data en total de las dos bases de datos existe una diferencia entre la cantidad de imágenes de personas sanas vs. las personas que presentan la enfermedad de Parkinson. Cabe mencionar que en el antecedente de investigación donde se habla de la utilización del *dataset* y la construcción del modelo de clasificación también se utilizó un *dataset* desbalanceado.

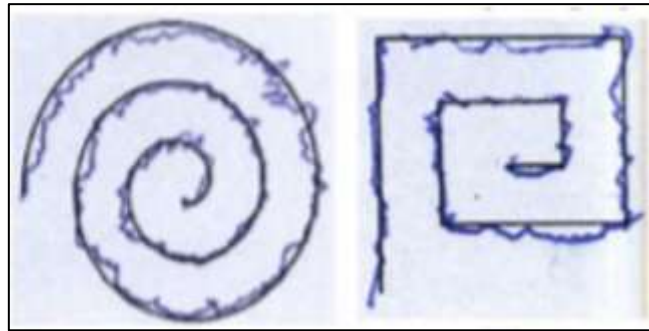


Figura 102. Ejemplos de espiral y meandro

Fuente: Clayton et al., (2016). *Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamics*. (p. 3)

Para el segundo *dataset*, se realizó una recolección de datos en la ciudad de Arequipa, es importante resaltar que estamos en tiempos de Pandemia por la COVID-19, por lo que se tuvieron que tomar medidas sanitarias para la recolección de datos. A continuación, se detalla la cronología de los eventos de la recolección de datos por fecha.

17 de agosto, 2020:

En este día se encontró una noticia que fue publicada el 23 de diciembre en el año 2019 por el diario Correo, titulada ¿Cómo superar el Parkinson? Consejos de una paciente que venció las dificultades (Figura 103). En el cual, se le hizo una entrevista a Christine Jeyachandran, donde ella explica que hacer deporte es bastante beneficioso para combatir la Enfermedad de Parkinson y que también tiene toda la disponibilidad para ayudar a otras personas que también presenten la misma o quieren ayudar a combatirla, es por eso que dejó el número de su celular disponible para conversar con ella.



Figura 103. Noticia diario Correo sobre EP

Fuente: Correo. (2019). *¿Cómo superar el Parkinson? Consejos de una paciente que venció las dificultades*.

18 de agosto, 2020: Primer contacto por Whatsapp

Christine es la fundadora de la Asociación “Actívate Arequipa Liga contra el Parkinson”, es por ello que el día 18 de agosto se contactó con Christine a través de una llamada y posteriormente por Whatsapp (Figura 104), donde se quedó en reunirnos en una conferencia virtual a través de la plataforma Zoom.

22 de agosto, 2020: Reunión a través de Zoom

En este día, se realizó la llamada por *Zoom*, donde se le detalló a Christine cual era el objetivo del presente trabajo de investigación y cómo se pretende realizar un pre-diagnóstico para detectar de manera anticipada la EP, asimismo, se detalló que se pretendía crear una base de datos similar a la que se creó en Brasil y por ello se necesitaba datos personales de los miembros de su Asociación para poder aplicarles el examen de escritura, ella comentó que le parecía una excelente ayuda y que estaba bastante interesada en apoyar con el desarrollo del trabajo.

27 de agosto, 2020: Carta de Presentación con los miembros de la Asociación

En este día, Christine mencionó que sería bueno elaborar una carta de presentación para que ella se la remita a los miembros de su Asociación por Whatsapp, para ver si ellos estaban interesados en apoyar. Se redactó la carta de presentación (Anexo 5) y se le envió a Christine. Ella comenta que no recibió respuesta de nadie, dado que la mayoría de las personas son de edad mayor y quizás no saben utilizar Whatsapp. Por lo cual, planteamos otra estrategia de tipo proactiva en la que se solicitó a Christine una lista con los datos de sus miembros.

03 de setiembre, 2020: Nombre y teléfono de 44 personas de la Asociación

Es por ello que, en este día ella envió un correo, en el cual se detalla el nombre y número de 44 personas que presentan la EP (Figura 105). Christine detalló, que ella no puede dar la dirección de las personas de su asociación y que primero se les tiene que pedir un permiso para poder ir a tomarles el examen.



Figura 104. Conversación con Christine por Whatsapp

Fuente: Elaboración propia



Figura 105. Correo sobre los nombres y apellidos de miembros de la Asociación

Fuente: Elaboración propia

06 y 07 de setiembre, 2020: Primer contacto por teléfono con las 44 personas, miembros de la Asociación

El domingo 6 y lunes 7 se procedió a llamar a esas 44 personas para contarles el objetivo del proyecto y asimismo solicitarles su apoyo para que participen en la recolección de datos. Después de hablar con las personas, muchos de ellos no querían que sus datos se vean expuestos por lo cual, se quedó en que el examen de escritura se desarrollaría de manera anónima. De las 44 personas, 27 dijeron que sí estaban interesadas en ayudar y debido a la pandemia que estamos atravesando se les planteó la idea de ir a sus domicilios el miércoles 9. Asimismo, se les indicó que ese día no se iba a ingresar a sus casas, que se tomarían los protocolos correspondientes, en los cuales tenía que haber dos metros de distancia con la puerta de sus casas y que el examen de escritura se encontraría dentro de un sobre manila encima de una mesa desplegable donde solo tenían que recogerlo y 3 días después este sería recogido, es decir el sábado 12.

08 de setiembre, 2020: Impresión del examen de escritura

El martes 8 se procedió a imprimir el examen de escritura (Anexo 6), instrucciones de llenado (Anexo 7), carta de presentación (Anexo 5) y un examen de escritura adicional en caso el primero sufriera algún tiempo de deterioro o contingencia. Los dos juegos de impresión del examen de escritura se colocaron dentro de un folder, la hoja de instrucciones fue pegada encima de la cara principal del folder, la carta de presentación estaba suelta, finalmente, el

folder y la carta de presentación fueron introducidos dentro de un sobre manila y este fue cerrado con cinta. En la Figura 106 se puede observar cómo fue todo el proceso para el armado de los sobres y en la Figura 107 como quedaron todos los sobres. Es importante resaltar que, a la hora de imprimir, anexar los documentos, entre otros, se tuvo el mayor de los cuidados utilizando alcohol, guantes y mascarilla.

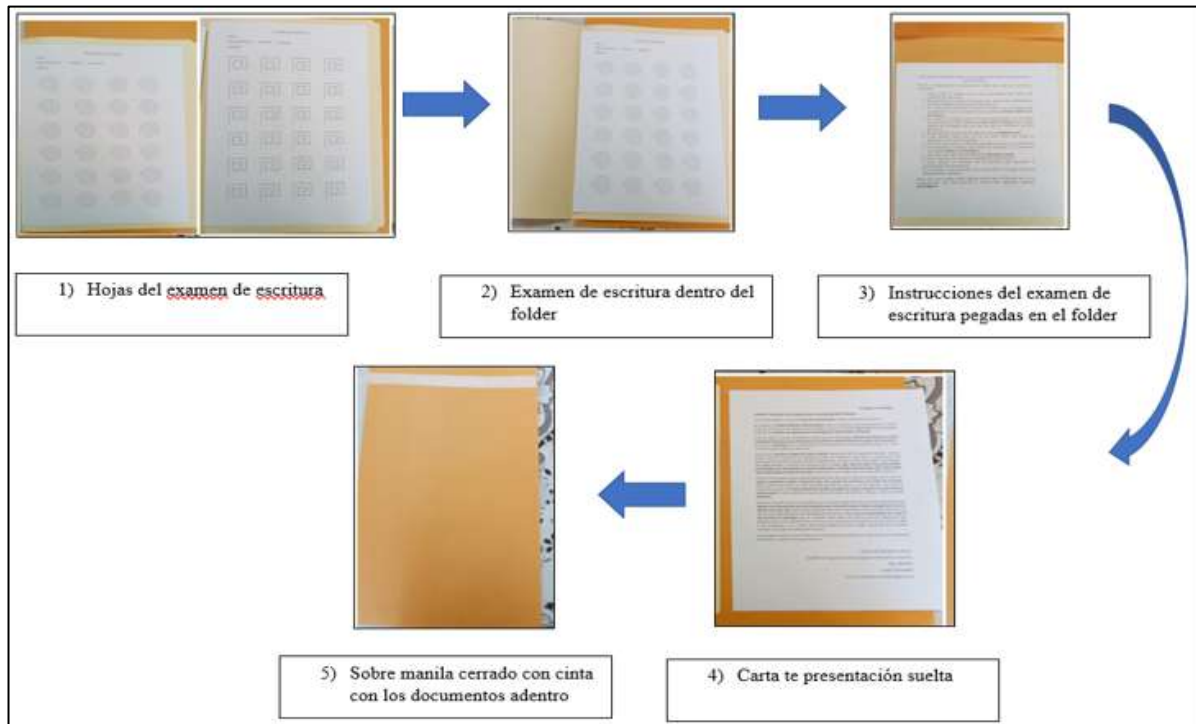


Figura 106. Proceso de armado de los sobres

Fuente: Elaboración propia



Figura 107. Fila de sobres finales

Fuente: Elaboración propia

09 de setiembre, 2020: Entrega de los sobres a las 27 personas, que presentan Enfermedad de Parkinson

El miércoles 9, se procedió a visitar todas las casas de las personas que habían confirmado su participación. Para empezar con el recorrido, se tomaron las medidas sanitarias correspondientes como el uso de los EPPs (Equipo de Protección Personal) el cual incluía, uso de mascarilla, protector facial y alcohol, asimismo, los sobres estaban dentro de una caja y en un vehículo personal. El recorrido se empezó temprano, llegando a la primera casa a las 06:53 am, una vez que se contactaba con la persona dueña de casa, se procedía a dejar en una mesita retráctil el sobre para que la persona pudiera recogerlo a 2 metros de distancia. En la Fig 104 se puede observar cómo fue el proceso de entrega de sobres. Ese día se hizo un recorrido por más de 10 distritos de la ciudad de Arequipa para poder entregar los sobres a todas personas, miembros de la Asociación. El recorrido terminó aproximadamente a las 3:00 pm y se le indicó a cada persona que tenía 3 días para el llenado de los exámenes de escritura, por lo cual el sábado 12 se procedería al recojo de los sobres.



Figura 108. Ejemplo de entrega de sobres

Fuente: Elaboración propia

12 de setiembre, 2020: Recojo de los sobres a las 27 personas, que presentan Enfermedad de Parkinson

El sábado 12, se procedió de misma forma a visitar todas las casas de las personas a las cuales se les entregó el sobre. Del mismo modo que para la entrega de los sobres se tomaron las medidas sanitarias correspondientes con el uso de los EPPs. El recorrido se empezó temprano, llegando a la primera casa a las 9:23 am, una vez que se contactaba con la persona dueña de casa, se procedía a dejar una caja encima de una mesita retráctil para que la persona pudiera dejar el sobre, respetando los 2 metros de distancia. En la Figura 109 y Figura 110 se puede observar cómo fue el proceso de recojo de sobres, cabe indicar que en la Figura 106 hay menos de 2 metros de distancia, ya que esa persona es Christine Jeyachandran, quien pidió una foto para que sea publicada en la revista de su Asociación (ver Anexo 8 para ver detalle). Ese día se hizo un recorrido por más de 10 distritos de la ciudad de Arequipa para poder recoger los sobres de todas personas, miembros de la Asociación. El recorrido terminó aproximadamente a las 3:20.



Figura 109. Ejemplo de recojo de sobres

Fuente: Elaboración propia



Figura 110. Recojo de sobre con Christine Jeyachandran

Fuente: Elaboración propia

19 y 20 de setiembre, 2020: Entrega y recojo de los sobres a las 27 personas, que no presentan la Enfermedad de Parkinson

Los días sábado y domingo se procedió con la entrega y recojo de los mismos sobres como en la Figura 108, que se les entregó a los miembros de la Asociación Actívate. Para este caso se seleccionaron 27 personas de Arequipa que no presentaban la Enfermedad de Parkinson. El proceso fue similar, sin embargo, en este caso se hizo la entrega y recojo de la prueba el mismo día, ya que al ser personas que no presentaban la EP se les hizo más sencillo el llenado del examen.

3 y 4 de octubre, 2020: Procesando los sobres

Según la OMS, la COVID-19 puede sobrevivir hasta 72 horas en plástico, 4 horas en superficies de cobre y menos de 24 horas en cartón. (OMS, 2020).

No especifican cuanto tiempo vive le virus en el papel, sin embargo, a fin de que las muestras recolectadas se desinfecten, se esperó dos semanas, tiempo que se consideró prudente, por si alguna de las personas a quienes se les aplicó el examen de escritura presentara la COVID-19. El sábado 3 se procedió con la desinfección con alcohol (Figura 109), revisión de los sobres y la selección de los exámenes de escritura.



Figura 111. Desinfección de los sobres

Fuente: Elaboración propia

El día 4 se procedió a realizar la toma de fotos para poder procesar los exámenes de escritura. Se buscó un lugar con una buena iluminación y se posicionó una plantilla de hoja en la pared para que todas las fotos sean tomadas en la misma posición, asimismo, se utilizó las

cámaras traseras de un celular Moto G9 Plus (Figura 112), cuyas características son las siguientes:

- Sensor principal: 64 Mpx.
- Ultra gran angular: 8 Mpx f/2.2.
- Macro: 2 Mpx.
- Sensor profundidad: 2Mpx.



Figura 112. Modelo de celular Moto G9 Plus que se utilizó para tomar foto de los exámenes de escritura

Fuente: Xataka (2020). *Motorola Moto G9 Plus: la "superbatería" llega al Moto G9 más potente hasta la fecha.*

y finalmente, se utilizó un trípode para que el celular siempre este en la misma posición y todas las fotos tengan las mismas medidas, que se posicionó a 13 cm del examen de escritura. (Figura 113).

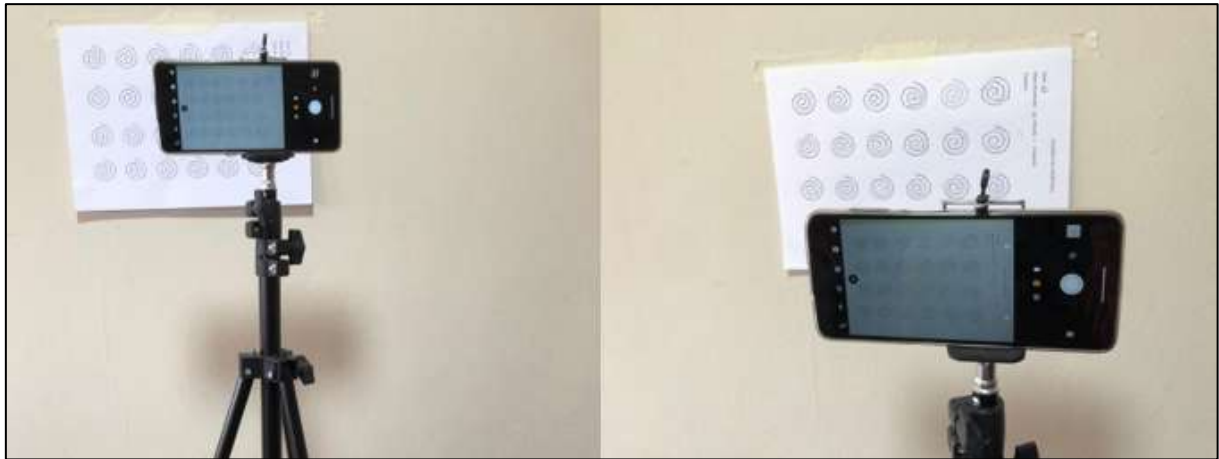


Figura 113. Trípode con el celular para tomar las fotos

Fuente: Elaboración propia

Finalmente, en la Figura 114 se ve un ejemplo de cómo quedó tomada la foto de un examen de escritura.

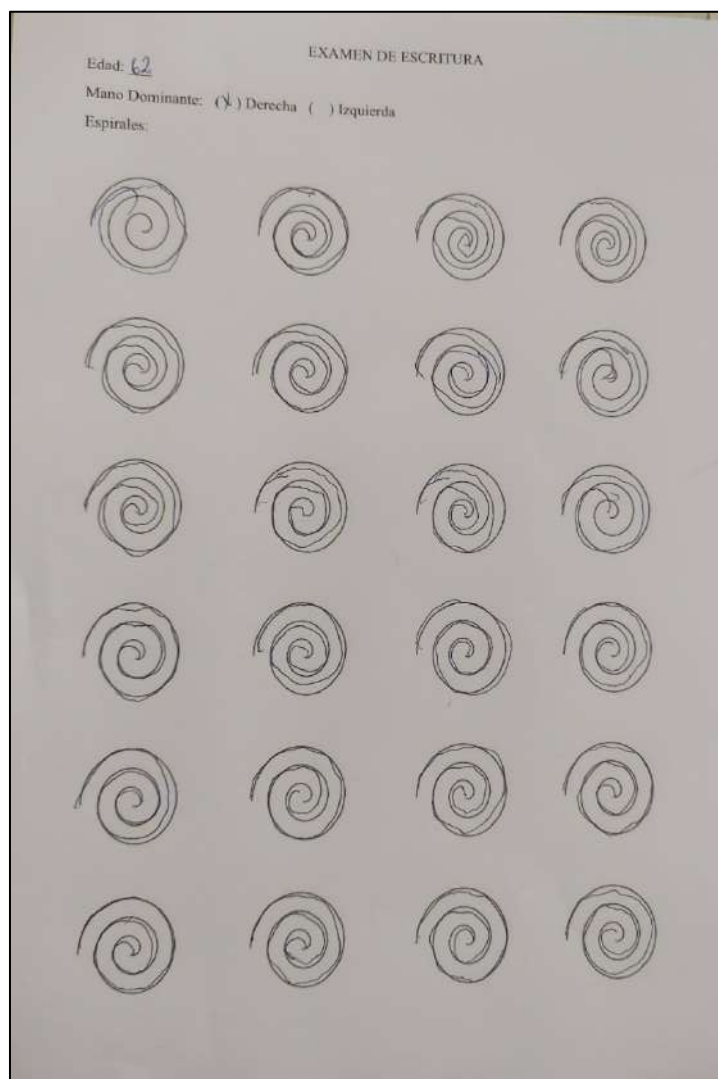


Figura 114. Ejemplo del examen de escritura

Fuente: Elaboración propia

4.2 Fase de Preprocesamiento

Para el primer *dataset*, como primer paso, se aplicó la técnica de *Data Augmentation* con el fin de poder crear una base de datos más amplia y así poder tener más imágenes para entrenar nuestros modelos de clasificación. Por cada imagen de la base de datos se han creado 5 adicionales con características como rotación de 90°, rotación de 180°, la función *flip* que es aplicarle un espejo a la imagen real, el nivel de brillo y se aplicaron filtros como la mediana. En la Figura 115 se puede observar algunos de los ejemplos de las nuevas imágenes generadas con *Data Augmentation*. Finalmente, en total la cantidad de imágenes en la nueva base de datos es 2940 de meandros y 1484 de espirales, es decir el volumen de imágenes se quintuplicó.

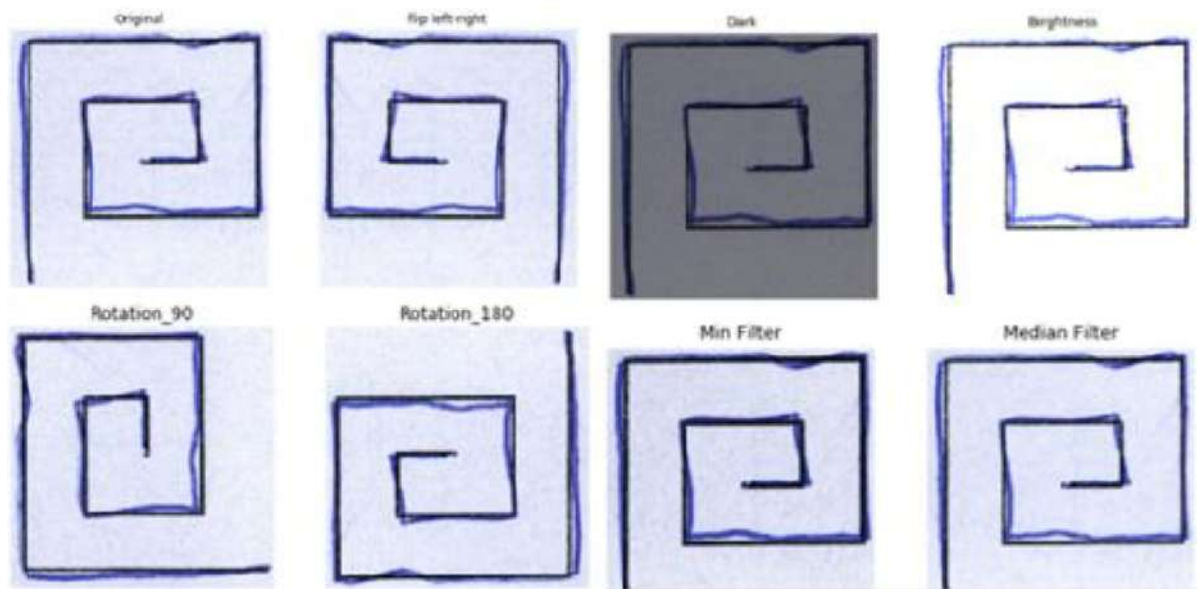


Figura 115. Ejemplos de imágenes generadas con Data Augmentation – Dataset Brasil

Fuente: Elaboración propia

Seguidamente se redimensionaron todas las imágenes con una dimensión estándar de 560x520, esta dimensión se determinó en base al promedio de las medidas de todas las imágenes. Una vez redimensionadas se procedió a pasar las imágenes a escala de grises, la cual se puede observar en la Figura 116, para poder aplicar tratamiento de imágenes para eliminar el ruido, aumento de intensidad, observar valores *outliers*, entre otros.

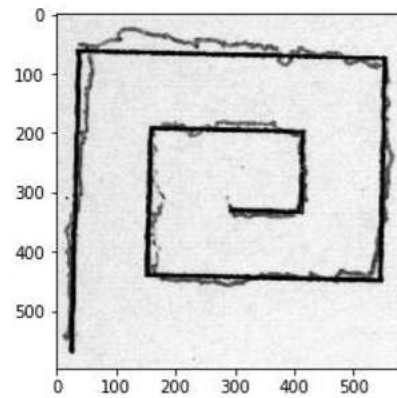


Figura 116. Imagen de un meandro en escala de grises

Fuente: Elaboración propia

Para el segundo *dataset*, como primer paso se elaboró un código en Python, llamado **Creación_base_de_datos.ipynb**, dado que todas las imágenes fueron tomadas desde la misma distancia, posición y entorno, se construyó un algoritmo el cual recorta cada figura dentro del examen de escritura en una misma posición, es decir se especificaron coordenadas de corte donde solo se pueda ver la figura de meandros o espirales. Una vez ejecutado el código se procedió a tener 25 imágenes de meandros y 20 imágenes de espirales por examen de escritura. En la Figura 118 se puede observar cómo quedaron estas imágenes.

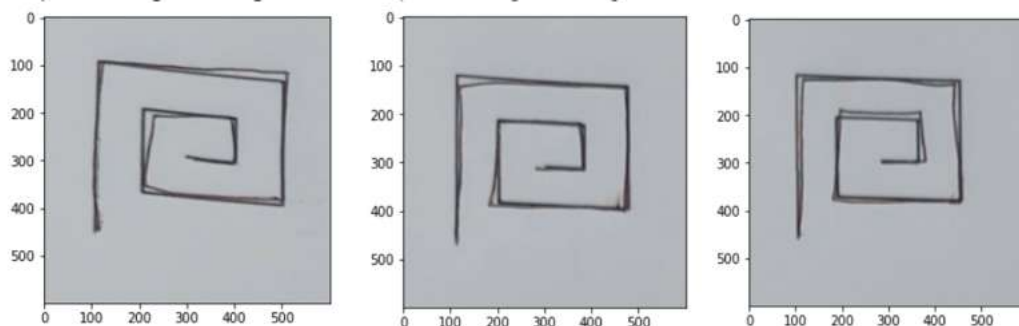


Figura 117. Ejemplos de imágenes que fueron cortadas del examen de escritura – Dataset Perú

Fuente: Elaboración propia

En total, se obtuvieron 1296 imágenes de meandros entre positivos y negativos y 1080 imágenes de espirales entre positivos y negativos. Es importante resaltar, que tanto para meandros como para espirales las clases se encuentran balanceadas. Seguidamente, se procedió a pasar la imagen a escala de grises.

Es importante resaltar en este punto que inicialmente, solo se aplicó escala de grises a la imagen como en la Figura 118. Sin embargo, después de realizar algunos experimentos de modelamiento, los resultados de la predicción del modelo evaluados por AUC no superaban el 50% en el conjunto de datos de prueba (*test*). Sin embargo, en el conjunto de datos de entrenamiento (*train*) se obtuvieron resultados de más del 80% de buena clasificación, lo cual indicaba *overffiting* (Figura 119), es decir que algoritmo no clasificaba de manera correcta a las imágenes nuevas y solo funcionaba con las imágenes que se usaron para entrenar el modelo. Por lo cual, se decidió aplicar técnicas de preprocesamiento a la imagen.

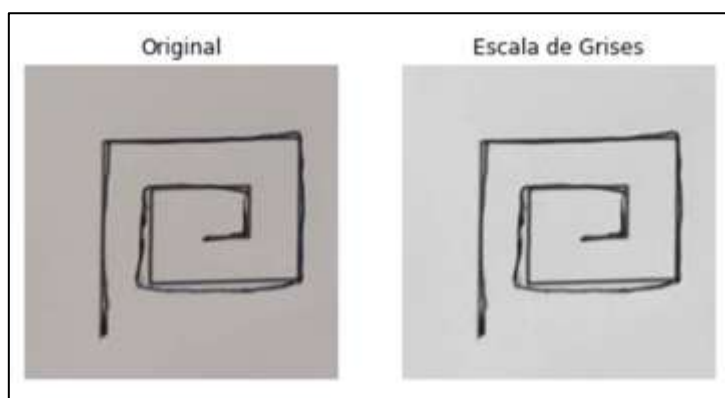


Figura 118. Primer preprocesamiento aplicado a las imágenes – Dataset Perú

Fuente: Elaboración propia

TIPO: 1	AUC TRAIN: 46.4%	AUC TEST: 33.7%

0.6390243902439025		
TIPO: 10	AUC TRAIN: 68.9%	AUC TEST: 29.6%

0.6231155778894473		
TIPO: 100	AUC TRAIN: 83.9%	AUC TEST: 26.5%

0.6102564102564103		
TIPO: 1000	AUC TRAIN: 98.2%	AUC TEST: 33.7%

0.6446700507614214		

Figura 119. Métricas obtenidas del primer preprocesamiento aplicado a las imágenes del segundo dataset

Fuente: Elaboración propia

Se procedió a escalar todas las imágenes a 600 x 600, ya que este era un promedio de tamaño de todas las imágenes. Una vez escaladas, se le aplicó un filtro de mediana con size=3, es decir la matriz de entrada para cada imagen fue de tamaño 3x3, aumento de contraste con enhance=2, lo cual significa que el aumento de contraste se aumentó en un 100% y brillo con

enhance=1.5, lo cual significa que la imagen tiene un ajuste del brillo en un 50%, ya que si el valor sería 1 el resultado sería la misma imagen. Este preprocesamiento se realizó para que la imagen se vea mucho mejor y el algoritmo pueda procesarla de una mejor forma. En la Figura 116 se puede observar como quedó la imagen después de todo el preprocesamiento aplicado.

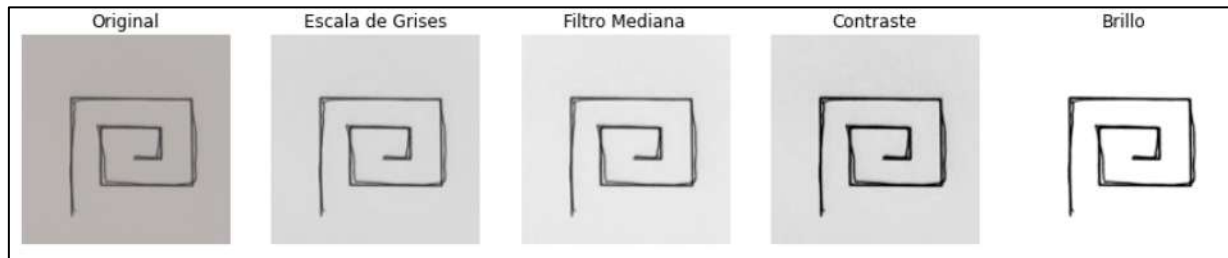


Figura 120. Segundo preprocesamiento aplicado a las imágenes – Dataset Perú

Fuente: Elaboración propia

Una vez preprocesadas las imágenes se aplicó la técnica de *Data Augmentation* para poder crear una base de datos con más imágenes y así la red pueda aprender de más patrones. Por cada imagen de la base de datos se han creado 5 adicionales con características como rotación de 90°, rotación de 180°, la función *flip* que es aplicarle un espejo a la imagen real. En la Figura 121 se puede observar los ejemplos de las nuevas imágenes generadas con *Data Augmentation*. Finalmente, en total la cantidad de imágenes en la nueva base de datos es 5400 de meandros y 6480 de espirales.

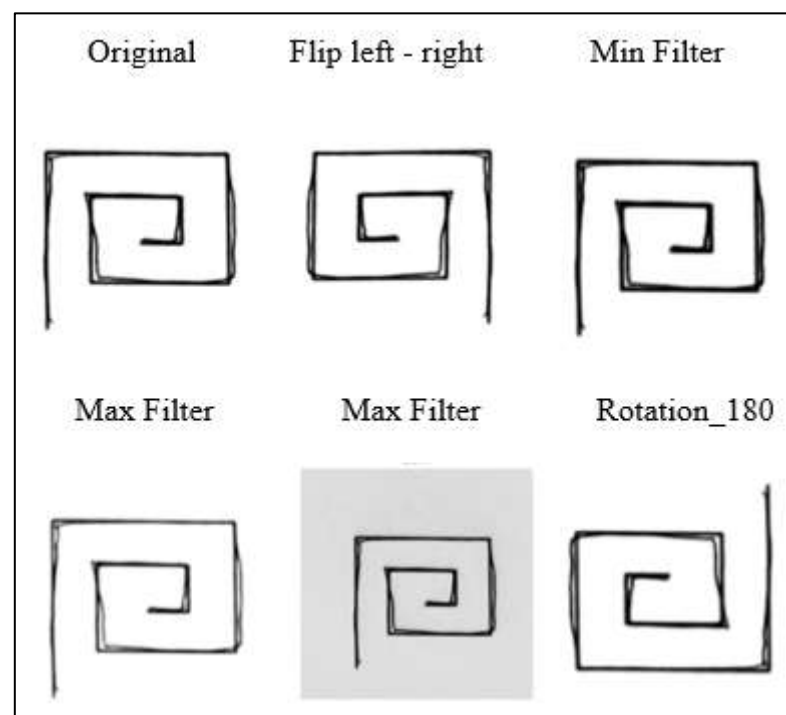


Figura 121. Ejemplos de imágenes generadas con Data Augmentation – Dataset Perú

Fuente: Elaboración propia

4.3 Extracción de características

En el presente punto, se utilizaron las técnicas de extracción de características SIFT, SURF y ORB con ayuda de la metodología BOVW y la técnica HOG para ambos *datasets*.

SIFT, SURF y ORB:

Para cada una de estas técnicas se utilizó BOVW y fueron probadas con diferentes tipos de K, este parámetro sirve para agrupar los distintos puntos característicos en clusters según sea el valor con el objetivo de formar una barra de histogramas y de ese modo poder obtener los vectores característicos, como K = 10, K = 50 y K = 100. Se construyeron dos funciones las cuales se pueden ver en la Figura 122 y Figura 123, la primera función *features*, es aquella que nos sirve para poder establecer que extractor de características se usó como SIFT, SURF o ORB y nos da como resultados los *keypoints* y los descriptores que en tamaño son 128 para SIFT, 64 para SURF y 32 para ORB.

```

1 def features(image, extractor):
2     keypoints, descriptors = extractor.detectAndCompute(image, None)
3     #print(descriptors)
4     #print("La longitud de los descriptors son:" + str(len(descriptors)))
5     return keypoints, descriptors

```

Figura 122. Función features

Fuente: Elaboración propia

En el caso de la función BOVW, esta es aquella que sirve para poder construir el diccionario y los histogramas para el BOVW, aquí se puede observar que se utiliza el *k-means* para poder agrupar los descriptores de las imágenes.


```

1 def BOVW(feature_descriptors, n_clusters):
2     cont = 0
3     print("Bag of visual words with {} clusters".format(n_clusters))
4     combined_features = np.vstack(np.array(feature_descriptors))
5     print("Empieza K-means TRAIN")
6     kmeans = KMeans(n_clusters=n_clusters, random_state=0).fit(combined_features)
7     print("Terminó K-means TRAIN")
8     bovw_vector = np.zeros([len(feature_descriptors), n_clusters])
9     for index, features in enumerate(feature_descriptors):
10        cont = cont + 1
11        print("Imagen" + str(cont+1))
12        try:
13            for i in kmeans.predict(features):
14                bovw_vector[index, i] += 1
15        except:
16            pass
17    return bovw_vector

```

Figura 123. Función BOVW

Fuente: Elaboración propia

Seguidamente, dado que ya se aplicó el BOVW tanto para espirales como para meandros, se procedió a normalizar los datos y a ponerlos en diferentes tablas. Como se puede observar en la Figura 124 y Figura 125.

```

1 normalizado_sift_meandros_total_10 = scaler.fit_transform(preprocessed_image_sift_meandros_total_10)
2 normalizado_sift_meandros_total_50 = scaler.fit_transform(preprocessed_image_sift_meandros_total_50)
3 normalizado_sift_meandros_total_100 = scaler.fit_transform(preprocessed_image_sift_meandros_total_100)
4 normalizado_sift_spiral_total_10 = scaler.fit_transform(preprocessed_image_sift_spiral_total_10)
5 normalizado_sift_spiral_total_50 = scaler.fit_transform(preprocessed_image_sift_spiral_total_50)
6 normalizado_sift_spiral_total_100 = scaler.fit_transform(preprocessed_image_sift_spiral_total_100)
7 normalizado_surf_meandros_total_10 = scaler.fit_transform(preprocessed_image_surf_meandros_total_10)
8 normalizado_surf_meandros_total_50 = scaler.fit_transform(preprocessed_image_surf_meandros_total_50)
9 normalizado_surf_meandros_total_100 = scaler.fit_transform(preprocessed_image_surf_meandros_total_100)
10 normalizado_surf_spiral_total_10 = scaler.fit_transform(preprocessed_image_surf_spiral_total_10)
11 normalizado_surf_spiral_total_50 = scaler.fit_transform(preprocessed_image_surf_spiral_total_50)
12 normalizado_surf_spiral_total_100 = scaler.fit_transform(preprocessed_image_surf_spiral_total_100)
13 normalizado_orb_meandros_total_10 = scaler.fit_transform(preprocessed_image_orb_meandros_total_10)
14 normalizado_orb_meandros_total_50 = scaler.fit_transform(preprocessed_image_orb_meandros_total_50)
15 normalizado_orb_meandros_total_100 = scaler.fit_transform(preprocessed_image_orb_meandros_total_100)
16 normalizado_orb_spiral_total_10 = scaler.fit_transform(preprocessed_image_orb_spiral_total_10)
17 normalizado_orb_spiral_total_50 = scaler.fit_transform(preprocessed_image_orb_spiral_total_50)
18 normalizado_orb_spiral_total_100 = scaler.fit_transform(preprocessed_image_orb_spiral_total_100)

```

Figura 124. Normalización de los datos

Fuente: Elaboración propia

	0	1	2	3	4	5	6	7	8	9
0	-0.184795	-0.634553	0.761159	-0.547708	0.746986	1.407038	-0.288783	-0.299186	0.656518	-0.250951
1	-0.211895	-0.353166	1.340805	-0.673998	1.663976	0.147333	0.288353	-0.064562	1.034654	-0.328159
2	-0.320295	-0.391537	2.186122	-0.126741	2.334084	0.738623	0.144069	0.035992	0.391822	-0.077232
3	-0.184795	0.247980	1.316653	0.841483	0.993868	0.687206	0.432637	0.136545	0.202753	0.057883
4	-0.157696	-0.608972	0.495488	-0.989723	0.641179	0.455832	-0.108428	0.069509	1.223723	-0.154440

Figura 125. Ejemplo de los datos normalizados

Fuente: Elaboración propia

HOG:

En el caso del HOG los vectores característicos son arrojados directamente después de utilizar la función HOG, lo que se realizó fue una selección de parámetros para reducir la cantidad de descriptores que bota el modelo HOG. En la Figura 128 se pueden observar los parámetros que se definieron para el HOG.

```
winSize = (64,64)
blockSize = (16,16)
blockStride = (8,8)
cellSize = (8,8)
nbins = 9
derivAperture = 1
winSigma = 4.
histogramNormType = 0
L2HysThreshold = 2.0000000000000001e-01
gammaCorrection = 0
nlevels = 64
```

Figura 126. Selección de parámetros

Fuente: Elaboración propia

En la Figura 127 se puede observar cómo se obtuvieron los vectores característicos.

```
hog = cv2.HOGDescriptor(winSize,blockSize,blockStride,cellSize,nbins,derivAperture,winSigma,
                        histogramNormType,L2HysThreshold,gammaCorrection,nlevels)
hog_meandros_sanos = []
cont = 0
winStride = (8,8)
padding = (8,8)
locations = ((10,20),)
for file in meandros_Sanos:
    image = Ruta_Meandros_Sanos + file
    #print(image)
    image = cv2.imread(image,cv2.IMREAD_GRAYSCALE)
    fv_hog = hog.compute(image,winStride,padding,locations)
    hog_meandros_sanos.append(fv_hog)
    cont = cont + 1
print(cont)
#break;
```

```
(hog_meandros_sanos[2])

array([[0.07161922],
       [0.07986903],
       [0.05733499],
       ...,
       [0.19200934],
       [0.16336933],
```

Figura 127. Obtención de los vectores característicos con HOG

Fuente: Elaboración propia

4.4 Modelo y Clasificación

Para el siguiente punto se debe tomar en cuenta que existen tres escenarios SVM, CNN y CNN + ML, los cuales van a ser detallados a continuación.

4.4.1 metodología 1

En este caso, dados los vectores característicos que se obtuvieron en la etapa de extracción de características, se procede a asignar el *target* correspondiente “1” si presentan la enfermedad de Parkinson y “0” si no presentan la enfermedad del Parkinson. En la Figura 124 se puede observar un ejemplo de una de las tablas con su respectivo *TARGET*.

1 tabla_sift_meandros_total_10_final.head()										
	0	1	2	3	4	5	6	7	8	9 TARGET
0	-0.184795	-0.634553	0.761159	-0.547708	0.746986	1.407038	-0.288783	-0.299186	0.656518	-0.250951 0
1	-0.211895	-0.353166	1.340805	-0.673998	1.663976	0.147333	0.288353	-0.064562	1.034654	-0.328159 0
2	-0.320295	-0.391537	2.186122	-0.126741	2.334084	0.738623	0.144069	0.035992	0.391822	-0.077232 0
3	-0.184795	0.247980	1.316653	0.841483	0.993868	0.687206	0.432637	0.136545	0.202753	0.057883 0
4	-0.157696	-0.608972	0.495488	-0.989723	0.641179	0.455832	-0.108428	0.069509	1.223723	-0.154440 0

Figura 128. Asignación del *TARGET* para los vectores característicos SVM

Fuente: Elaboración propia

Seguidamente, se hizo un *shuffle* para que los datos en cada tabla no estén en el mismo orden según *target* y como se puede observar se le asigna una semilla lo cual indica que se está trabajando con el mismo *shuffle* para todas las tablas.

1 # Shuffle										
2 from sklearn.utils import shuffle										
3 tabla_sift_meandros_total_10_final = shuffle(tabla_sift_meandros_total_10_final, random_state = 123)										
4 tabla_sift_meandros_total_10_final.head()										
	0	1	2	3	4	5	6	7	8	9 TARGET
662	1.603797	2.307225	-0.905323	2.209625	0.605910	-0.238291	0.793347	0.840418	-0.175383	0.887875 1
172	-0.238995	-0.058988	1.413261	0.167936	1.452363	-0.958123	0.216211	-0.064562	0.618704	-0.096534 0
174	-0.293195	-0.046197	1.389109	0.210032	2.228278	-0.855290	-0.216641	-0.198633	0.807772	-0.038627 0
493	-0.293195	0.554948	-0.615500	-0.337225	-0.416886	-0.212583	-0.577351	-0.466775	-0.969471	-0.366764 1
496	-0.374495	1.143304	-0.301525	0.630999	-0.593230	-0.084042	-0.072357	0.136545	-0.213197	0.173696 1

Figura 129. Función *shuffle* para los vectores característicos

Fuente: Elaboración propia

Finalmente, se procedió a dividir los datos en *train* (70%), *test* (20%) y *val* (10%), y se aplicó el modelo SVM con *GridSearchCV* para poder elegir los mejores parámetros para cada conjunto de datos. El análisis de los resultados será expuesto en el siguiente Capítulo.

```

1 from sklearn.model_selection import GridSearchCV
2
3 def train_test_svm_grid(tabla):
4     params = {'kernel':('linear','rbf','sigmoid'),'C':[1,10,100,1000],'gamma':[0.1,0.01,0.001,0.0001]}
5
6     feature_cols = list(set(tabla.columns.tolist()) - set(['TARGET']))
7     X = tabla[feature_cols]
8     y = tabla.TARGET
9
10    train_ratio = 0.70
11    validation_ratio = 0.10
12    test_ratio = 0.20
13
14    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1 - train_ratio, random_state=123)
15    X_val, X_test, y_val, y_test = train_test_split(X, y, test_size=test_ratio/(test_ratio + validation_ratio),
16                                                    random_state=123)
17
18    #print(X_train, X_val, X_test)
19
20    clf = svm.SVC()
21    clf = GridSearchCV(clf, params, cv=5)
22    clf.fit(X_train, y_train)
23    print(clf.best_params_)

```

Figura 130. Función para particionar los datos y aplicar SVM

Fuente: Elaboración propia

```

C:\Users\USER\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:841: DeprecationWarning: The default of the 'iid'
parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results when tes
t-set sizes are unequal.
  DeprecationWarning)
{'C': 100, 'gamma': 0.01, 'kernel': 'rbf'}

```

Figura 131. Aplicación de GridSearchCV con SVM

Fuente: Elaboración propia

4.4.2 metodología 2

En este caso primero empezaré indicando los tipos de estructuras que se utilizaron para estos experimentos, los cuales fueron VGG16, VGG19, Inception, Resnet y Lenet.

Del mismo modo las imágenes que se utilizaron ya estaban divididas en *train* (70%), *test* (20%) y *val* (10%). Las imágenes fueron escaladas y al momento de utilizar las arquitecturas se utilizaron los pesos de un modelo que ya fue entrenado llamado ImageNet, asimismo, las imágenes están 256x256x3.

```
## load the base model of VGG-16 without the last layer
base_model = VGG16(weights="imagenet", #los valores del kernel son los Ws
                    include_top=False, #solamente llamamos parte de la GG16, hasta Max pooling
                    input_shape = (256, 256, 3)
                    )

## architecture summary
base_model.summary()
```

Figura 132. Cargando el modelo VGG16

Fuente: Elaboración propia

Asimismo, en todas las arquitecturas se le agregaron dos capas más de las que tenía y finalmente se le agregó la salida con una función Sigmoidal.

```
## add custom layers
X = base_model.output #cap1
X = Flatten()(X) #cap2
X = Dense(1024, activation="relu")(X) ## 250,088 + 1,024 + 1,024 #capa3
X = Dropout(rate=0.5)(X) #capa4
X = Dense(1024, activation="relu")(X) #capa5

## add the last layer with softmax activation for 3 classes
#output = Dense(2, activation="softmax")(X) ## for categorical
output = Dense(1,
               activation='sigmoid')(X) ## for binary #capa6

## build the model
model = Model(input = base_model.input, output = output)

## freeze:
## the vgg layers
## the first 10 layers
## not all
for layer in model.layers[:]: #[:-5] solo entrena las 5 últimas capa de vgg
    layer.trainable = True

## architecture summary
model.summary()
```

Figura 133. Agregación de dos capas a la arquitectura de VGG16

Fuente: Elaboración propia

En la Figura 130 se puede observar el código para compilar y entrenar una red.

```

## compile the model
# model.compile(optimizer = optimizers.SGD(lr=0.00001, momentum=0.7),
#               loss = "categorical_crossentropy",
#               metrics = ["accuracy"])
model.compile(optimizer= optimizers.SGD(lr=INIT_LR, momentum=0.9),#Stocastid Gradient Descent
              loss='binary_crossentropy',#
              metrics=['accuracy'])

## get the number of image by each batch
#step_size_train = train_generator.n // train_generator.batch_size
#step_validation_steps = test_generator.n // test_generator.batch_size

## train the model
H = model.fit_generator(aug.flow(trainX, trainY, batch_size= len(trainX) // BS),
                      validation_data=(testX, testY), steps_per_epoch=len(trainX) // BS,
                      epochs=EPOCHS)

```

Figura 134. Código para compilar y entrenar una red

Fuente: Elaboración propia

Finalmente, se procedió a entrenar el modelo y la métrica con la que se evaluó cada época fue Accuracy. Asimismo, los diferentes parámetros que se utilizaron para entrenar las redes se muestran en las Tablas 20 y 21 para meandros y espirales respectivamente de la base de datos de Brasil y para la base de datos de Perú se muestran en las Tablas 22 y 23, cabe indicar que los parámetros que se muestran son aquellos con los que se obtuvo mejor rendimiento. Es importante mencionar que se hicieron diversos experimentos hasta encontrar los más estables y con mejores métricas de evaluación.

Base de Datos Brasil:

Meandros:

Tabla 20.

Parámetros de las redes para meandros – Dataset Brasil

	Learning_Rate	Epochs	Batch_size
VGG16	0.001	100	64
VGG19	0.0001	100	128
Inception	0.01	100	64
ResNet	0.001	100	256
LeNet	0.0001	100	64

Fuente: Elaboración propia

Espirales:

Tabla 21.

Parámetros de las redes para espirales – Dataset Brasil

	Learning_Rate	Epochs	Batch_size
VGG16	0.01	100	64
VGG19	0.001	100	64
Inception	0.001	100	64
ResNet	0.01	100	256
LeNet	0.0001	100	128

Fuente: Elaboración propia

Base de Datos Perú:

Meandros:

Tabla 22.

Parámetros de las redes para meandros – Dataset Perú

	Learning_Rate	Epochs	Batch_size
VGG16	0.01	100	64
VGG19	0.001	100	64
Inception	0.001	100	64
ResNet	0.001	100	256
LeNet	0.001	100	128

Fuente: Elaboración propia

Espirales:

Tabla 23.

Parámetros de las redes para espirales – Dataset Perú

	Learning_Rate	Epochs	Batch_size
VGG16	0.01	100	128
VGG19	0.001	100	128
Inception	0.01	100	64
ResNet	0.001	100	64

LeNet	0.01	100	128
-------	------	-----	-----

Fuente: Elaboración propia

4.4.3 metodología 3

En el caso de CNN + ML lo que se realizó fue extraer la penúltima capa de la red, es decir obtener los datos de la mencionada capa antes de que sea clasificada.

```
model_feat = Model(inputs=predictor_meandros_vgg16.input, outputs=predictor_meandros_vgg16.get_layer('dense_2').output)
```

Figura 135. Extracción de la penúltima capa de la red

Fuente: Elaboración propia

Seguidamente, se procedió a juntar todos los vectores característicos en una tabla y adjuntarle su TARGET correspondiente 1 si presenta la EP y 0 si era un paciente sano. Además, se aplicó la función *Shuffle* para alterar el orden de los datos. Finalmente, se aplicó SVM + GridSearchCV, RF + GridSearchCV y KNN + GridSearchCV, para poder sacar los mejores parámetros. Los resultados se podrán observar en el siguiente Capítulo.

```
tabla_train_total = pd.concat([df_train_parkinson[:], df_train_sanos[:]], axis=0, ignore_index=True)
tabla_train_total.head()
```

	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	TARGET
0	2.270532	0.000000	3.100332	0.0	2.469709	0.062557	0.0	0.0	0.0	0.712076	2.300922	0.000000	0.996825	0.099626	0.0	0.668123	2.122826	1
1	1.194136	0.000000	2.954023	0.0	0.923998	2.116536	0.0	0.0	0.0	1.234485	2.411304	0.541440	1.949836	0.000000	0.0	3.496005	2.927829	1
2	1.918283	1.399609	0.000000	0.0	0.235764	0.000000	0.0	0.0	0.0	0.000000	0.987070	2.111596	0.000000	0.314451	0.0	0.363458	0.729454	1
3	1.200189	0.064323	0.125076	0.0	1.200292	0.000000	0.0	0.0	0.0	0.000000	2.681124	0.265741	0.000000	0.572520	0.0	0.574726	2.194922	1
4	4.552793	0.000000	2.809620	0.0	0.000000	0.000000	0.0	0.0	0.0	1.363377	2.459887	0.000000	0.724599	0.077900	0.0	0.379805	3.367864	1

Figura 136. Construcción de la tabla de CNN + ML

Fuente: Elaboración propia

```
train_test_svm_grid(df_train_parkinson)
{'C': 1, 'gamma': 0.1, 'kernel': 'linear'}
```

Figura 137. Aplicación de GridSearchCV

Fuente: Elaboración propia

Asimismo, los parámetros con los cuales se crearon las funciones `train_test_svm_grid`, `train_test_rf_grid` y `train_test_knn_grid` para SVM, RF y KNN respectivamente, se pueden observar en las siguientes tablas.

SVM:

Tabla 24.

Parámetros para GridSearchCV - SVM

Parámetros	Valores			
kernel	linear	rbf	sigmoid	-
C	1	10	100	1000
gamma	0.1	0.01	0.001	0.0001

Fuente: Elaboración propia

RF:

Tabla 25.

Parámetros para GridSearchCV - RF

Parámetros	Valores				
n_estimators	200	500	-	-	-
max_features	auto	sqrt	log2	-	-
max_depth	4	5	6	7	8
criterion	gini	entropy	-	-	-

Fuente: Elaboración propia

KNN:

Tabla 26.

Parámetros para GridSearchCV - KNN

Parámetros	Valores			
n_neighbors	3	5	11	19
weights	uniform	distance	-	-
metric	euclidean	manhattan	-	-

Fuente: Elaboración propia

Capítulo 5: Análisis y Discusión de Resultados

A continuación, en el presente capítulo se muestran los resultados obtenidos en los modelos de extractores de características, CNN y CNN + ML tanto para meandros como para espirales. Los resultados están divididos en dos, primero los resultados de la base de datos de Brasil y posteriormente la de Perú.

5.1 Resultados Base de Datos de Brasil

- Metodología 1:

Respecto a SVM aplicado con los vectores característicos de SIFT, SURF y ORB, se obtuvieron los siguientes resultados con $k = 10$, $k = 50$ y $k = 100$:

- $k = 10$

Tabla 27.

Resultados con $k = 10$ – Dataset Brasil

		SIFT	SURF	ORB
MEANDROS				
	ACCURACY	0.85	0.85	0.74
	AUC_TRAIN	0.77	0.67	0.55
	AUC_TEST	0.76	0.81	0.52
	AUC_VAL	0.75	0.84	0.54
ESPIRALES				
	ACCURACY	0.85	0.86	0.75
	AUC_TRAIN	0.74	0.84	0.54
	AUC_TEST	0.76	0.79	0.57
	AUC_VAL	0.76	0.81	0.52

Fuente: Elaboración propia

- $k = 50$

Tabla 28.

Resultados con $k = 50$ – Dataset Brasil

		SIFT	SURF	ORB
MEANDROS				
	ACCURACY	0.95	0.92	0.78
	AUC_TRAIN	0.96	0.96	0.73
	AUC_TEST	0.95	0.85	0.61
	AUC_VAL	0.93	0.86	0.72
ESPIRALES				
	ACCURACY	0.91	0.89	0.83

	AUC_TRAIN	0.98	0.96	0.75
	AUC_TEST	0.90	0.86	0.72
	AUC_VAL	0.93	0.91	0.73

Fuente: Elaboración propia

- k = 100

Tabla 29.

Resultados con k = 100 – Dataset Brasil

		SIFT	SURF	ORB
MEANDROS				
	ACCURACY	0.95	0.88	0.80
	AUC_TRAIN	0.98	0.93	0.88
	AUC_TEST	0.89	0.80	0.69
	AUC_VAL	0.95	0.85	0.61
ESPIRALES				
	ACCURACY	0.90	0.88	0.79
	AUC_TRAIN	0.95	0.99	0.90
	AUC_TEST	0.83	0.82	0.66
	AUC_VAL	0.93	0.91	0.73

Fuente: Elaboración propia

Por lo tanto, se puede inferir que de los resultados en base a los descriptores SIFT, SURF y ORB con k = 10, para meandros el mejor modelo se obtuvo con SIFT y para espirales el mejor modelo se obtuvo con SURF. Con k = 50, para meandros el mejor modelo se obtuvo con SIFT y para espirales el mejor modelo se obtuvo también con SIFT. Además, con k=100, para meandros el mejor modelo se obtuvo con SIFT y para espirales también se obtuvo con SIFT.

Finalmente, analizando todas las métricas, se escoge como mejor modelo SIFT con k = 50 para meandros con un 95% de *Accuracy* y SIFT con k=100 para espirales con un 90% de *Accuracy*, dado que estos son los más estables en todas las métricas.

Respecto a SVM aplicado a HOG, se obtuvieron los siguientes resultados:

Tabla 30.

Resultados con HOG – Dataset Brasil

	HOG - Accuracy
MEANDROS	0.88
ESPIRAL	0.78

Fuente: Elaboración propia

Por lo cual, los modelos obtenidos por HOG se compararon en base a la métrica *Accuracy* y se determinó que el mejor resultado para meandros fue con un 88% de *Accuracy* y para espirales fue con un 0.78% de *Accuracy*. Es importante resaltar, que estos resultados son inferiores al mejor modelo escogido en base a descriptores característicos.

- Metodología 2:

Respecto a las redes neuronales convolucionales se obtuvieron los siguientes resultados, tanto para meandros como para espirales.

Meandros:

Tabla 31.

Resultados con CNN (meandros) – Dataset Brasil

	Accuracy	Precision	Recall	F1-Score
VGG16	1.00	0.98	1.00	1.00
VGG19	1.00	0.99	1.00	0.99
ResNet	0.53	0.53	1.00	0.69
Inception	0.93	0.97	0.89	0.93
LeNet	0.80	0.72	0.93	0.81

Fuente: Elaboración propia

Las siguientes gráficas muestran cómo es que las redes se entrenaron y sus resultados respecto al *Accuracy* y *Loss*.

VGG16

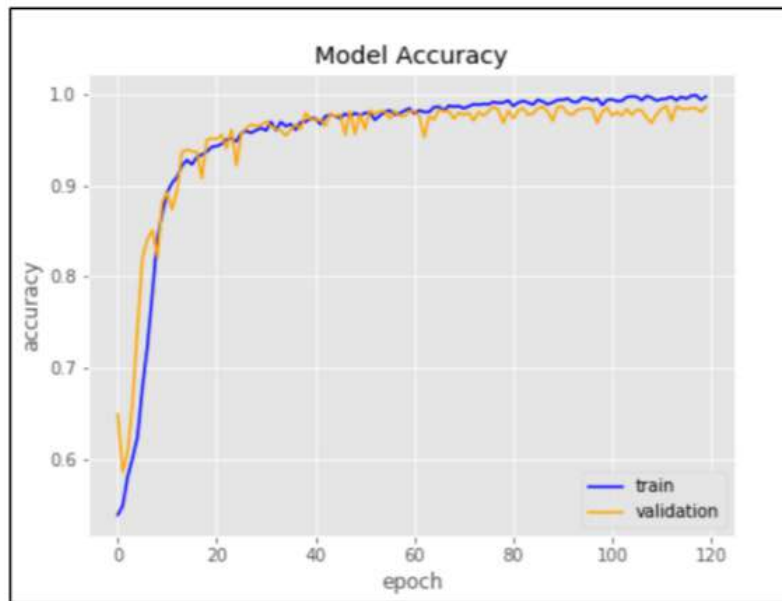


Figura 138. Gráfica de *Accuracy* VGG16 (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

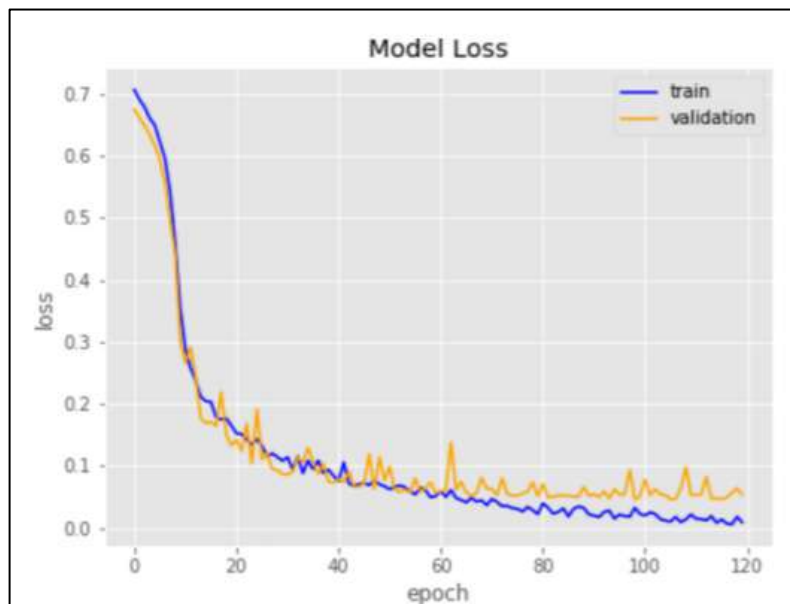


Figura 139. Gráfica de *Loss* VGG16 (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

VGG19

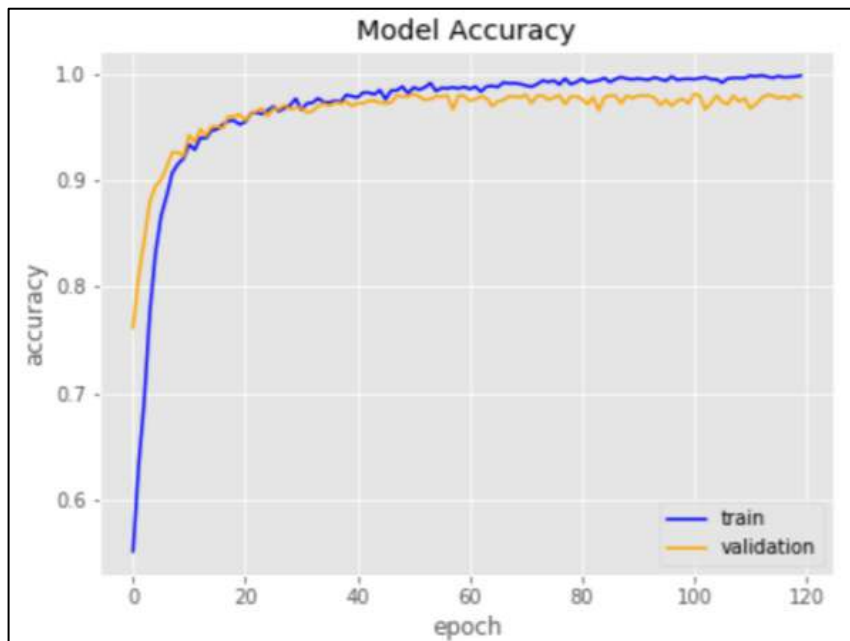


Figura 140. Gráfica de *Accuracy* VGG19 (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

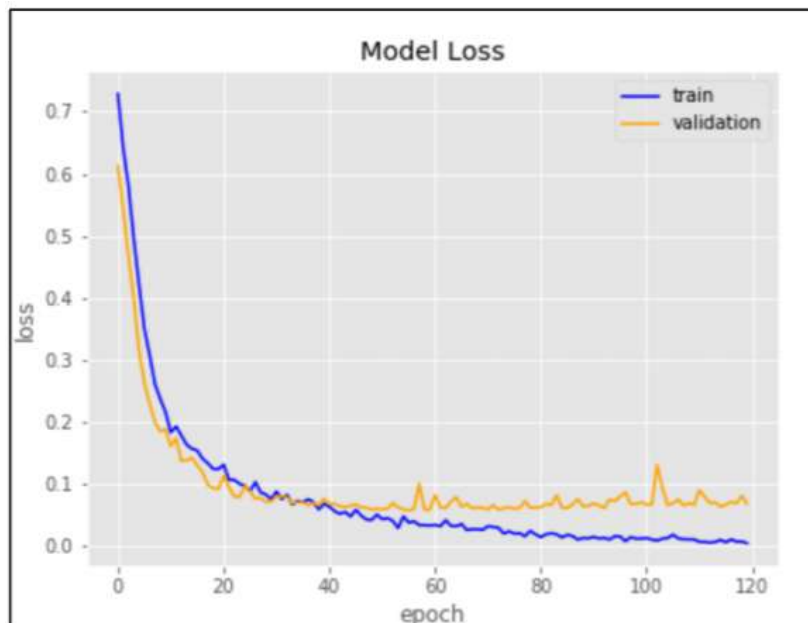


Figura 141. Gráfica de *Loss* VGG19 (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

Inception

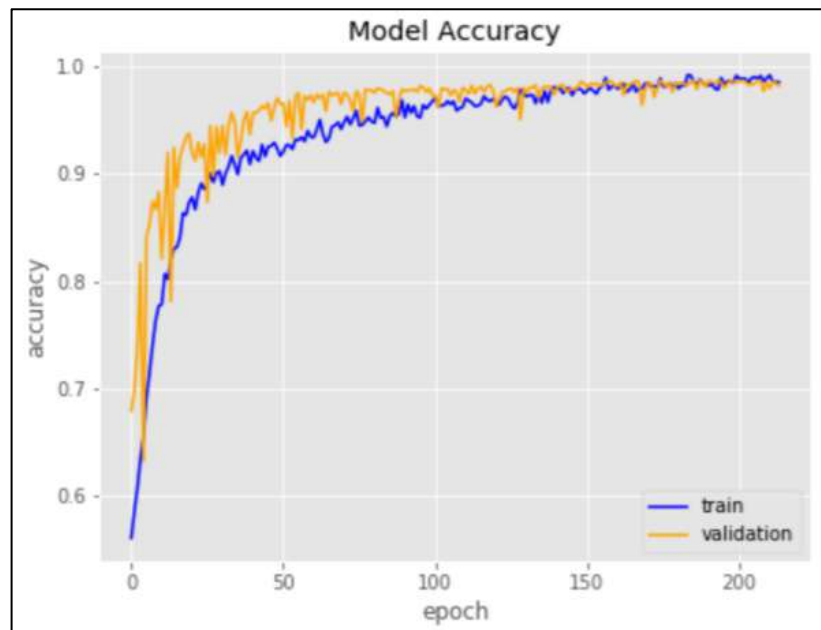


Figura 142. Gráfica de *Accuracy* Inception (meandros) – *Dataset* Brasil

Fuente: Elaboración propia

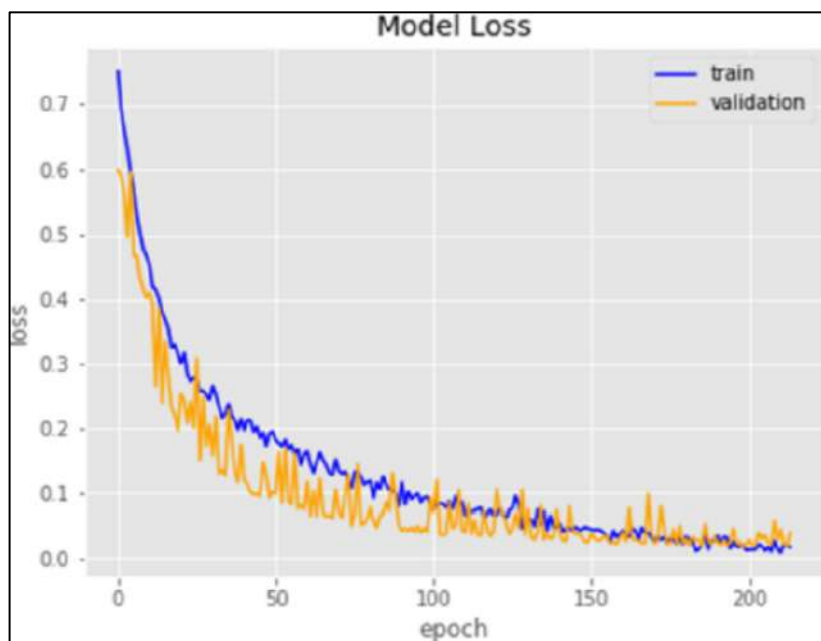


Figura 143. Gráfica de *Loss* Inception (meandros) – *Dataset* Brasil

Fuente: Elaboración propia

ResNet

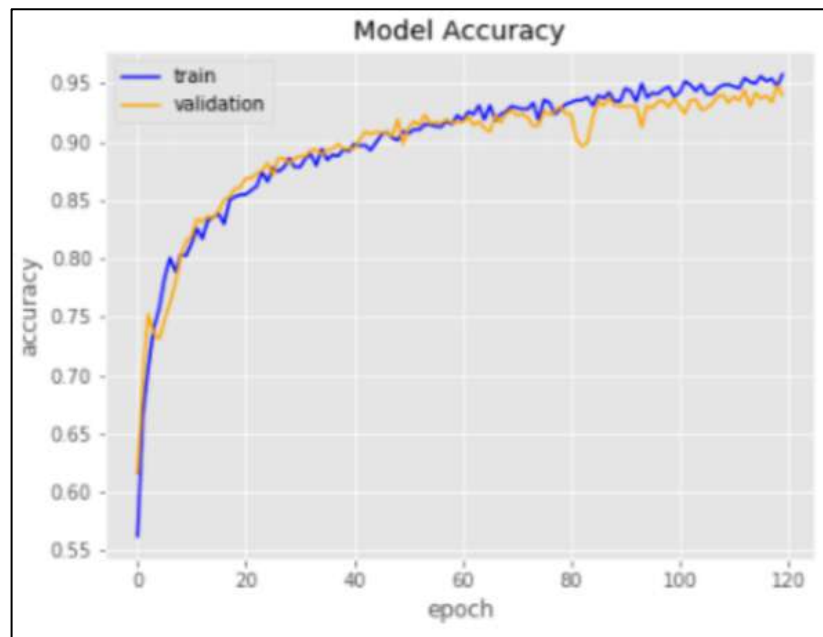


Figura 144. Gráfica de *Accuracy* ResNet (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

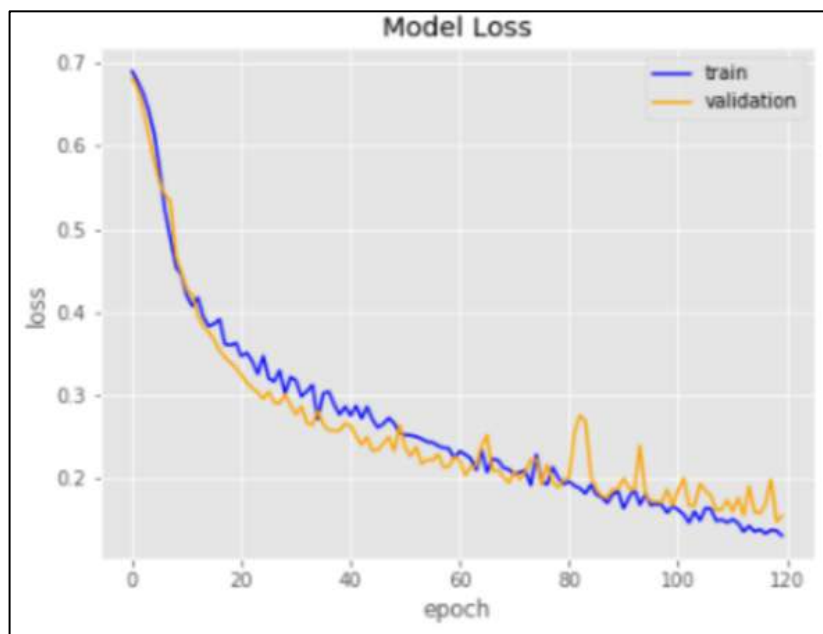


Figura 145. Gráfica de *Loss* ResNet (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

LeNet

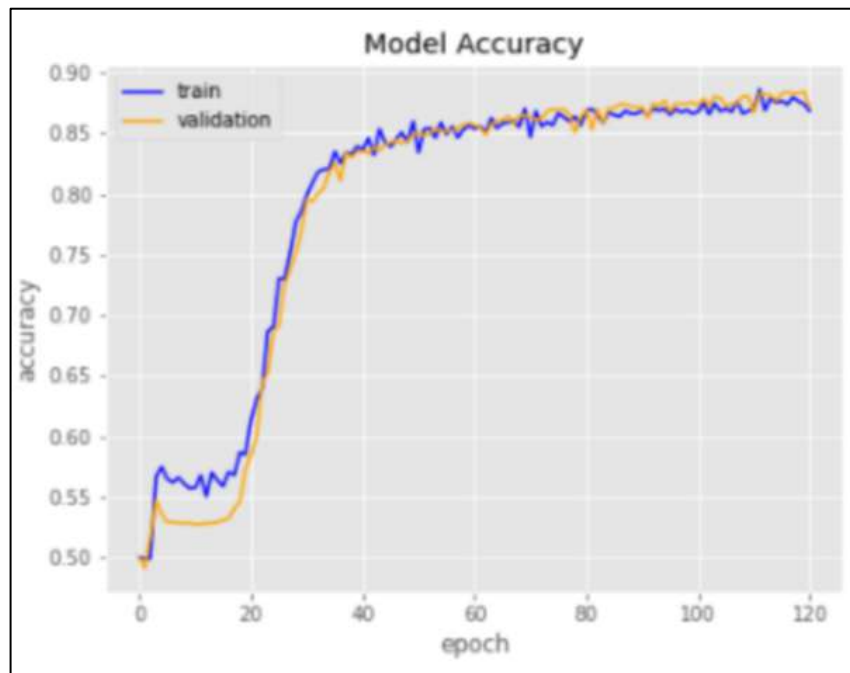


Figura 146. Gráfica de *Accuracy LeNet* (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

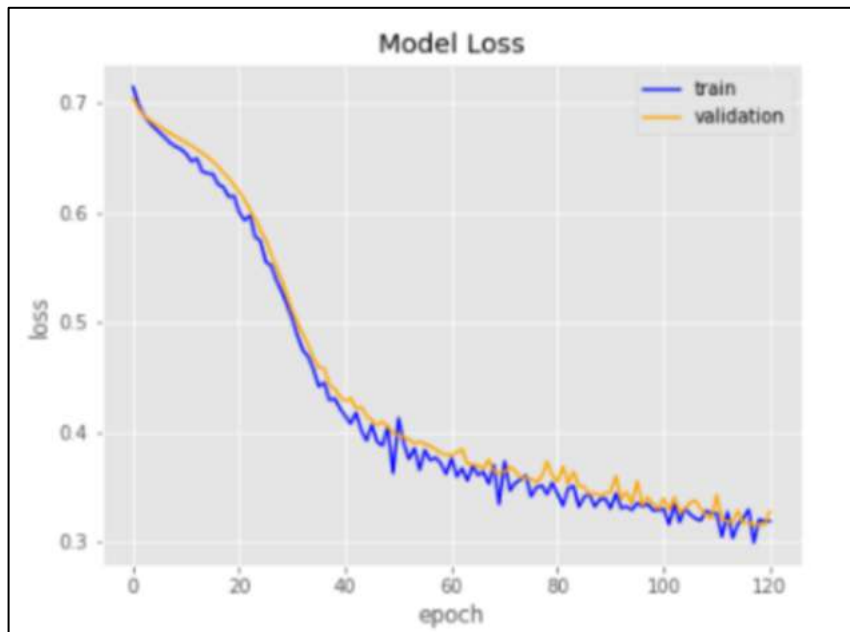


Figura 147. Gráfica de *Loss LeNet* (meandros) – *Dataset Brasil*

Fuente: Elaboración propia

Por lo tanto, se puede inferir que los mejores resultados para meandros se obtuvieron de la red VGG19, con 99.6% de *Accuracy*, cuya configuración fue `learning_rate = 0.0001`,

epochs=100 y batch_size =128. Además, la red que obtuvo menores resultados fue ResNet50, esto debido a la gran cantidad de capas que posee la red, por lo cual el modelo presenta overfitting.

Espirales:

Tabla 32.

Resultados con CNN (Espirales) – Dataset Brasil

	Accuracy	Precision	Recall	F1-Score
VGG16	1.00	0.98	0.99	1.00
VGG19	0.93	0.95	0.90	0.93
ResNet	0.50	0.49	1.00	0.66
Inception	0.89	0.89	0.88	0.89
LeNet	0.80	0.72	0.93	0.81

Fuente: Elaboración propia

Las siguientes gráficas muestran cómo es que las redes se entrenaron y sus resultados respecto al *Accuracy* y *Loss*.

VGG16

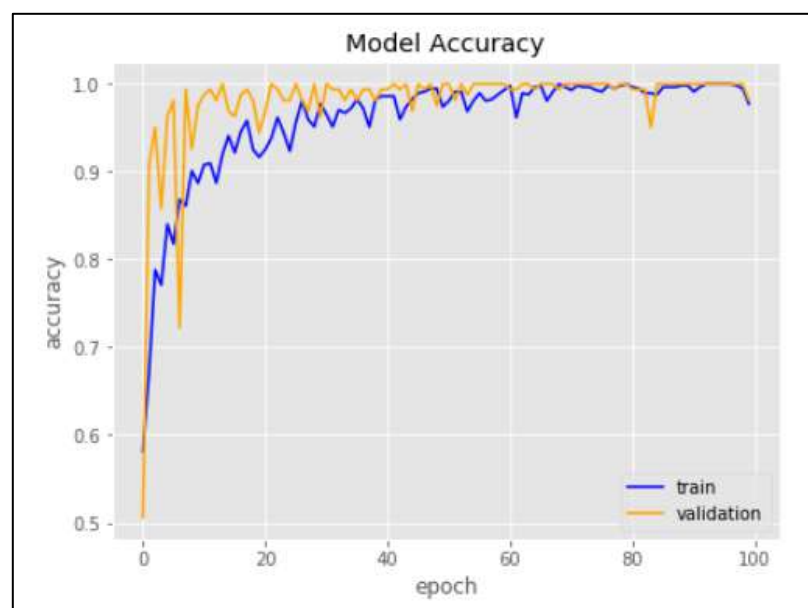


Figura 148. Gráfica de *Accuracy* VGG16 (espirales) – Dataset Brasil

Fuente: Elaboración propia

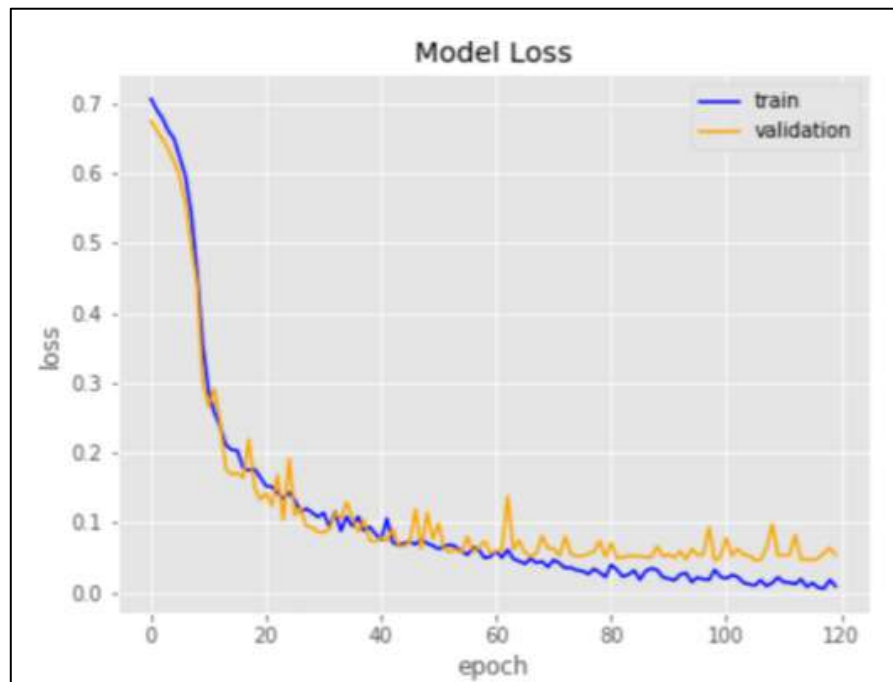


Figura 149. Gráfica de *Loss* VGG16 (espirales) – *Dataset* Brasil

Fuente: Elaboración propia

VGG19

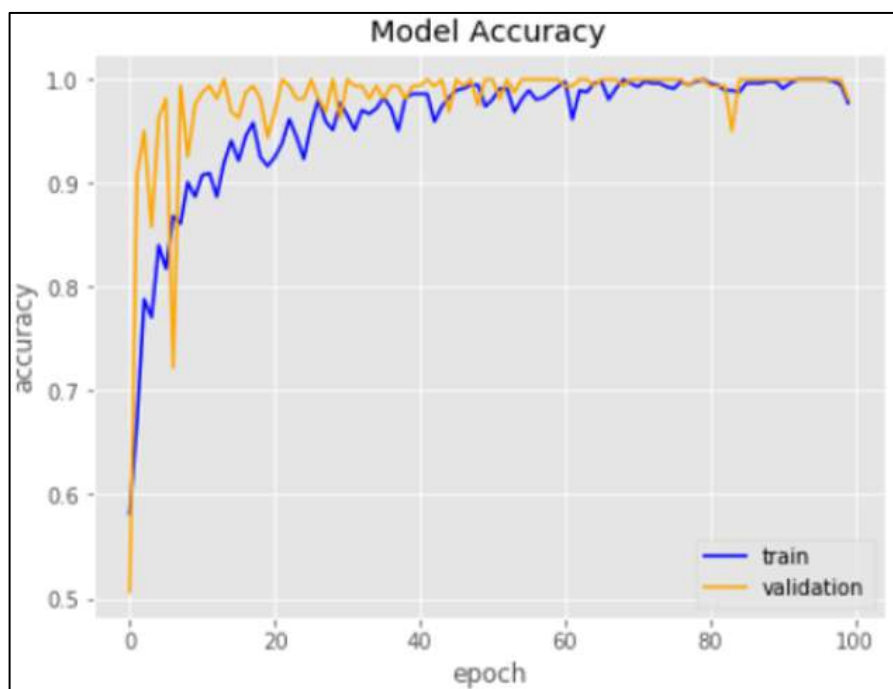


Figura 150. Gráfica de *Accuracy* VGG19 (espirales) – *Dataset* Brasil

Fuente: Elaboración propia

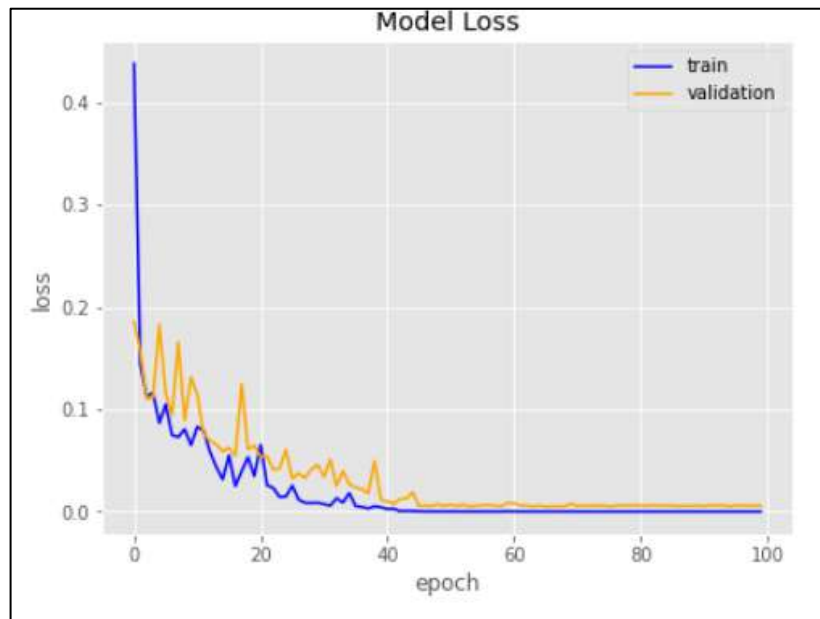


Figura 151. Gráfica de *Loss* VGG19 (espirales) – *Dataset* Brasil

Fuente: Elaboración propia

Inception



Figura 152. Gráfica de *Accuracy* Inception (espirales) – *Dataset* Brasil

Fuente: Elaboración propia

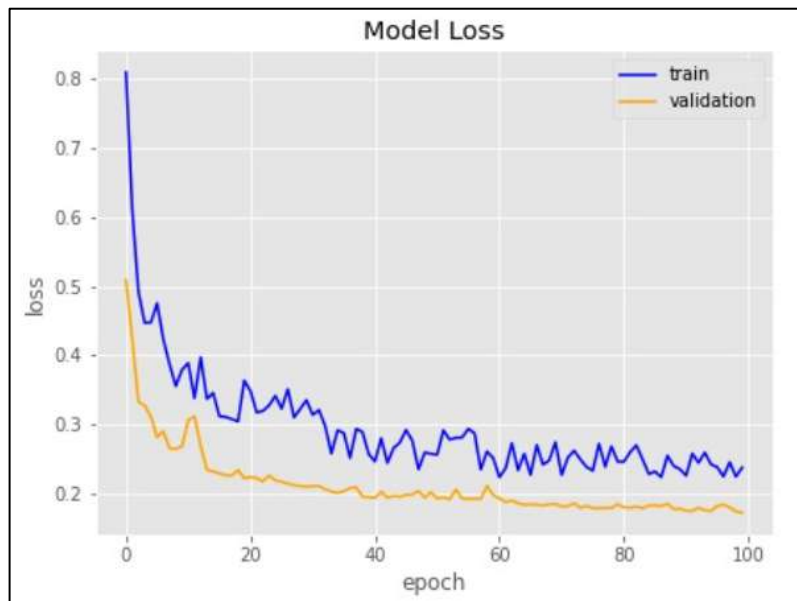


Figura 153. Gráfica de *Loss Inception* (espirales) – *Dataset Brasil*

Fuente: Elaboración propia

ResNet

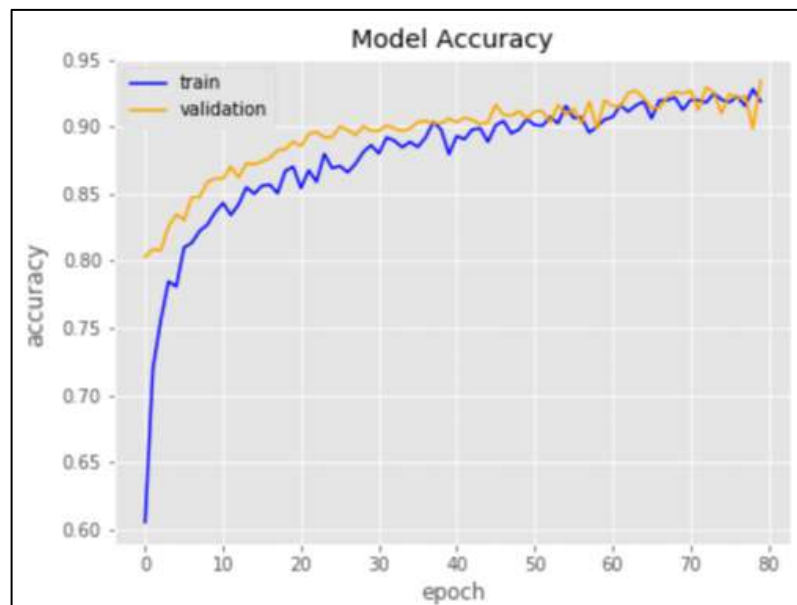


Figura 154. Gráfica de *Accuracy ResNet* (espirales) – *Dataset Brasil*

Fuente: Elaboración propia

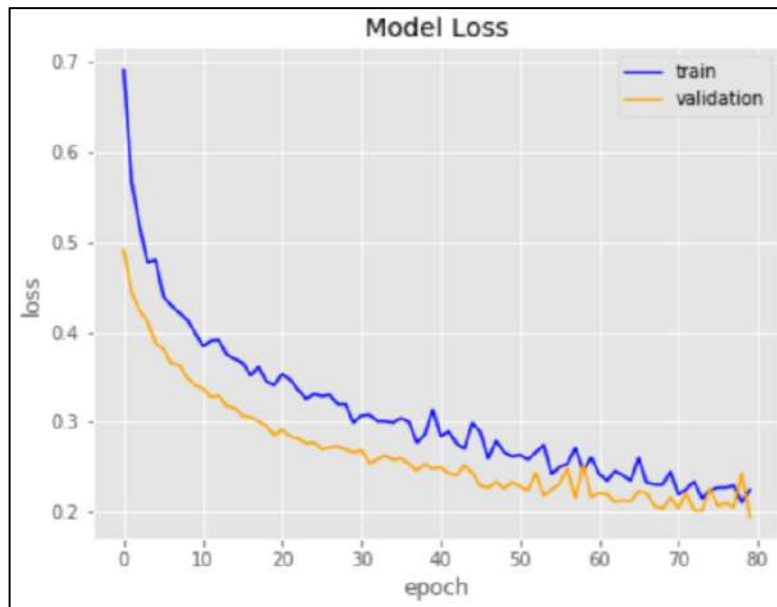


Figura 155. Gráfica de *Loss* ResNet (espirales) – *Dataset* Brasil

Fuente: Elaboración propia

LeNet

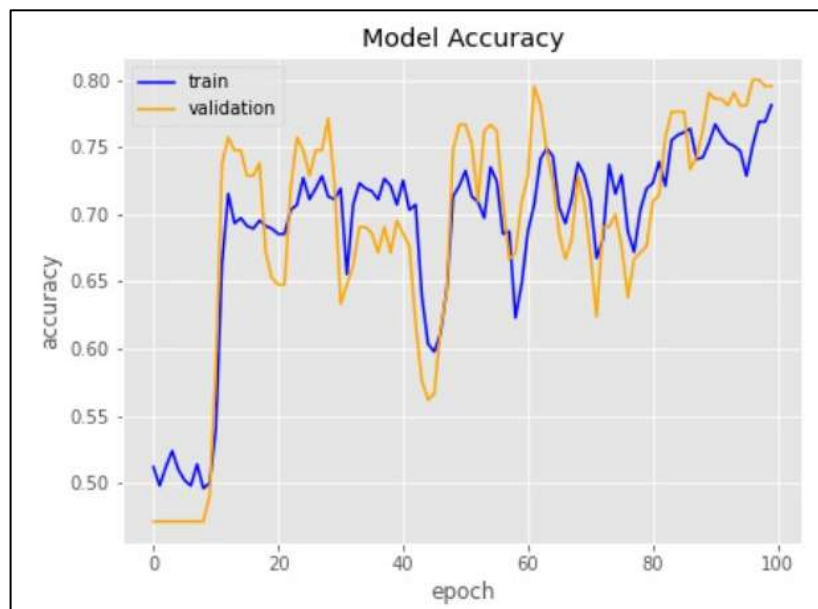


Figura 156. Gráfica de *Accuracy* LeNet (espirales) – *Dataset* Brasil

Fuente: Elaboración propia

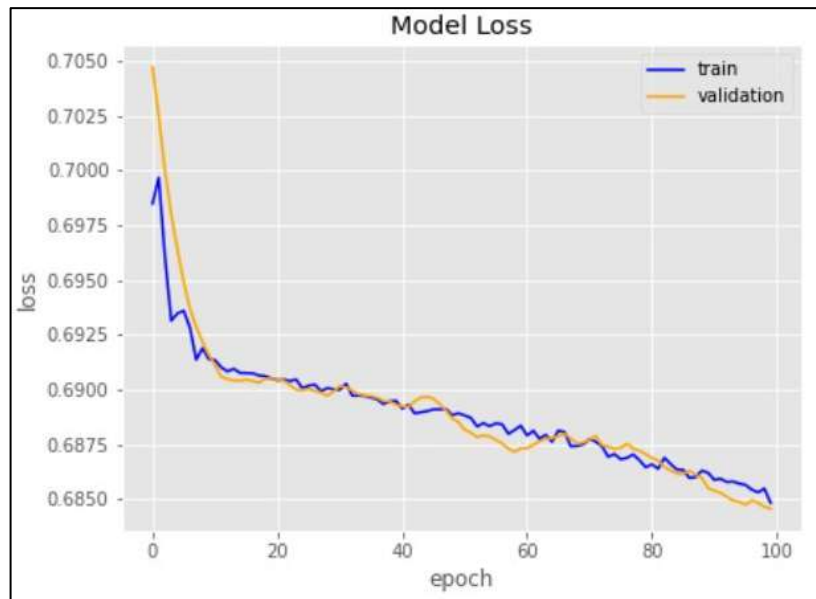


Figura 157. Gráfica de *Loss* LeNet (espirales) – *Dataset* Brasil

Fuente: Elaboración propia

Por lo tanto, se puede inferir que para espirales los mejores resultados se obtuvieron de la red VGG16, con 99.7% de *Accuracy*, cuya configuración fue *learning_rate* = 0.01, *epochs*=100 y *batch_size* =64. Además, la red que obtuvo menores resultados fue ResNet50, esto debido a la gran cantidad de capas que posee la red, por lo cual el modelo presenta overfitting.

- Metodología 3:

Respecto a las redes neuronales convolucionales con ML se obtuvieron los siguientes resultados, tanto para meandros como para espirales. Es importante resaltar que para cada algoritmo: SVM, RF y KNN se utilizó GridSearch con el objetivo de obtener los resultados óptimos. En la Figura 132 se puede observar un ejemplo de las configuraciones que resultan de aplicar GridSearch a SVM, RF y KNN. La matriz de confusión para cada experimento de meandros y espirales se puede observar en el Anexo 12.

```
SVM:  {'C': 1, 'gamma': 0.1, 'kernel': 'linear'}
RF     {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'n_estimators': 200}
KNN    {'metric': 'euclidean', 'n_neighbors': 11, 'weights': 'uniform'}
```

Figura 158. Ejemplo de GridSearchCV para SVM, RF y KNN

Fuente: Elaboración propia

MEANDROS:

Tabla 33.

Resultados con CNN + ML (meandros) – Dataset Brasil

		Accuracy	Precision	Recall	F1-Score	AUC
VGG16	SVM	0.99	0.99	0.97	0.98	0.98
	RF	0.98	0.96	0.98	0.97	0.97
	KNN	0.99	0.97	0.99	0.98	0.98
VGG19	SVM	0.72	0.84	0.68	0.75	0.77
	RF	0.91	0.89	0.92	0.91	0.9
	KNN	0.95	0.93	0.95	0.94	0.94
ResNet	SVM	0.98	0.99	0.99	0.99	0.99
	RF	0.98	0.97	0.99	0.98	0.98
	KNN	0.98	0.97	0.99	0.98	0.97
Inception	SVM	0.92	0.88	0.92	0.90	0.9
	RF	0.94	0.90	0.94	0.92	0.92
	KNN	0.97	0.97	0.98	0.97	0.97
LeNet	SVM	0.90	0.78	0.92	0.84	0.83
	RF	0.94	0.90	0.95	0.92	0.92
	KNN	0.97	0.93	0.97	0.95	0.94

Fuente: Elaboración propia

Se infiere, que los mejores resultados para meandros han sido obtenidos de la penúltima capa de la red VGG16, con 512 columnas de vectores característicos, a los cuales se le agregó la columna *target* y finalmente se aplicó algoritmos de ML, de los cuales se obtuvo el mejor resultado 99.8% de *Accuracy* con SVM, cuya configuración es de kernel='linear', C=1 y gamma=0.1.

ESPIRALES:

Tabla 34.

Resultados con CNN + ML (espirales) – Dataset Brasil

		Accuracy	Precision	Recall	F1-Score	AUC
VGG16	SVM	0.99	0.99	0.99	0.95	0.97
	RF	0.98	0.99	0.97	0.96	0.96

	KNN	1.00	1.00	1.00	1.00	0.98
VGG19	SVM	1.00	0.96	1.00	0.98	0.97
	RF	1.00	1.00	1.00	1.00	0.99
	KNN	1.00	1.00	1.00	1.00	0.99
ResNet	SVM	0.99	0.88	0.99	0.93	0.93
	RF	1.00	0.97	1.00	0.99	0.98
	KNN	0.99	0.97	1.00	0.98	0.98
Inception	SVM	0.92	0.88	0.95	0.91	0.93
	RF	0.98	0.96	0.98	0.97	0.97
	KNN	0.99	0.97	0.99	0.98	0.98
LeNet	SVM	0.92	0.80	0.94	0.86	0.84
	RF	0.95	0.90	0.96	0.93	0.92
	KNN	0.98	0.92	0.99	0.95	0.94

Fuente: Elaboración propia

Se infiere, que los mejores resultados para meandros han sido obtenidos de la penúltima capa de la red VGG19, con 512 columnas de vectores característicos, a los cuales se le agregó la columna *target* y finalmente se aplicó algoritmos de ML, de los cuales se obtuvo el mejor resultado 99.8% de *Accuracy* con KNN, cuya configuración es de *metric=euclidean*, *n_neighbors=3* y *weights='uniform'*.

Con todos los resultados mostrados anteriormente, se puede concluir que, para la base de datos de Brasil, los mejores resultados salen de las combinaciones de CNN + SVM tanto para meandros como para espirales con 99.8% de *Accuracy* en ambos modelos.

5.2 Resultados Base de Datos de Perú

- Metodología 1:

Respecto a SVM aplicado con los vectores característicos de SIFT, SURF y ORB, se obtuvieron los siguientes resultados con $k = 10$, $k = 50$ y $k = 100$:

- $k = 10$

Tabla 35.

Resultados con $k = 10$ – Dataset Perú

		SIFT	SURF	ORB
--	--	------	------	-----

MEANDROS				
	ACCURACY	0.81	0.71	0.81
	AUC_TRAIN	0.92	0.70	0.89
	AUC_TEST	0.80	0.69	0.80
	AUC_VAL	0.91	0.69	0.89
ESPIRALES				
	ACCURACY	0.82	0.83	0.77
	AUC_TRAIN	0.84	0.89	0.82
	AUC_TEST	0.78	0.81	0.75
	AUC_VAL	0.85	0.90	0.83

Fuente: Elaboración propia

- k = 50

Tabla 36.

Resultados con k = 50 – Dataset Perú

		SIFT	SURF	ORB
MEANDROS				
	ACCURACY	0.85	0.95	0.90
	AUC_TRAIN	0.96	1.00	1.00
	AUC_TEST	0.85	0.89	0.89
	AUC_VAL	0.96	1.00	1.00
ESPIRALES				
	ACCURACY	0.93	0.94	0.81
	AUC_TRAIN	1.00	0.99	0.83
	AUC_TEST	0.86	0.88	0.77
	AUC_VAL	1.00	0.99	0.82

Fuente: Elaboración propia

- k = 100

Tabla 37.

Resultados con k = 100 – Dataset Perú

		SIFT	SURF	ORB
MEANDROS				

	ACCURACY	0.99	0.91	0.93
	AUC_TRAIN	1.00	1.00	1.00
	AUC_TEST	0.84	0.89	0.92
	AUC_VAL	1.00	1.00	1.00
ESPIRALES				
	ACCURACY	0.90	0.80	0.89
	AUC_TRAIN	1.00	0.84	0.99
	AUC_TEST	0.89	0.78	0.87
	AUC_VAL	1.00	0.85	0.99

Fuente: Elaboración propia

Por lo tanto, se puede inferir que de los resultados en base a los descriptores SIFT, SURF y ORB con $k = 10$, para meandros el mejor modelo se obtuvo con SIFT y para espirales el mejor modelo se obtuvo con SURF. Con $k = 50$, para meandros el mejor modelo se obtuvo con SIFT y para espirales el mejor modelo se obtuvo con SURF. Además, con $k=100$, para meandros el mejor modelo se obtuvo con ORB y para espirales también se obtuvo con ORB.

Finalmente, analizando todas las métricas, se escoge como mejor modelo SIFT con $k = 50$ para meandros con un 85% de *Accuracy* y SURF con $k=100$ para espirales con un 94% de *Accuracy*, dado que estos son los más estables en todas las métricas.

Respecto a SVM aplicado a HOG, se obtuvieron los siguientes resultados:

Tabla 38.

Resultados con HOG – Dataset Perú

	HOG - Accuracy
MEANDROS	0.85
ESPIRAL	0.73

Fuente: Elaboración propia

Por lo cual, los modelos obtenidos por HOG se compararon en base a la métrica *Accuracy* y se determinó que el mejor resultado para meandros fue con un 85% de *Accuracy* y para espirales fue con un 0.73% de *Accuracy*. Es importante resaltar, que estos resultados son inferiores al mejor modelo escogido en base a descriptores característicos.

- Metodología 2:

Respecto a las redes neuronales convolucionales se obtuvieron los siguientes resultados, tanto para meandros como para espirales.

Meandros:

Tabla 39.

Resultados con CNN (meandros) – Dataset Perú

	Accuracy	Precision	Recall	F1-Score
VGG16	0.99	0.99	1.00	0.99
VGG19	1.00	1.00	0.99	0.99
ResNet	0.99	0.99	0.99	0.99
Inception	0.76	0.79	0.71	0.75
LeNet	0.96	1.00	0.93	0.96

Fuente: Elaboración propia

Las siguientes gráficas muestran cómo es que las redes se entrenaron y sus resultados respecto al *Accuracy*.

VGG16

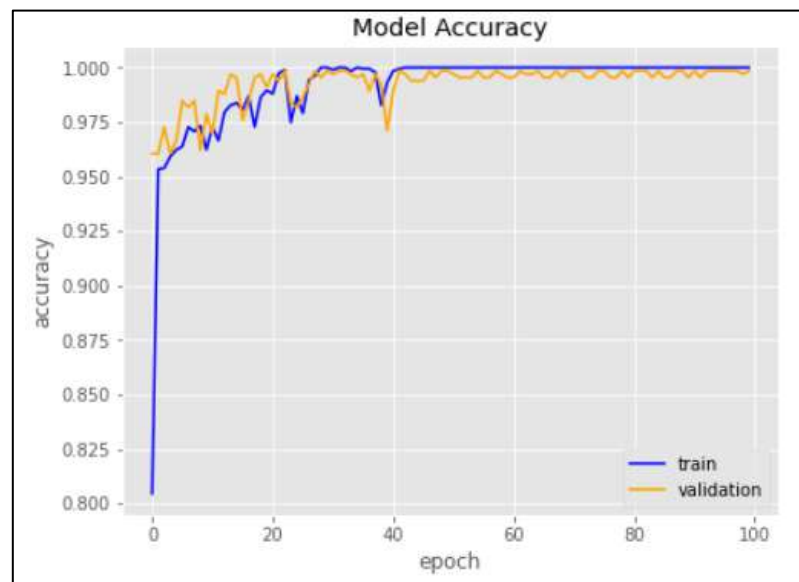


Figura 159. Gráfica de *Accuracy* VGG16 (meandros) – Dataset Perú

Fuente: Elaboración propia

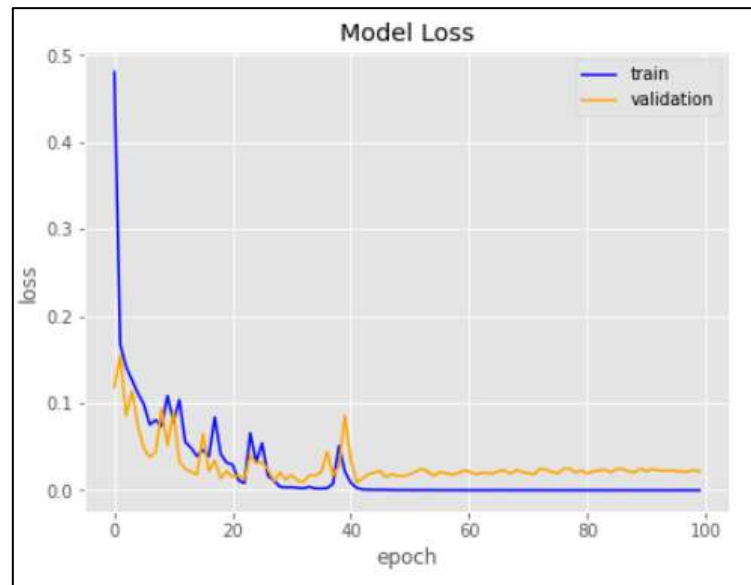


Figura 160. Gráfica de *Loss VGG16* (meandros) – *Dataset Perú*

Fuente: Elaboración propia

VGG19

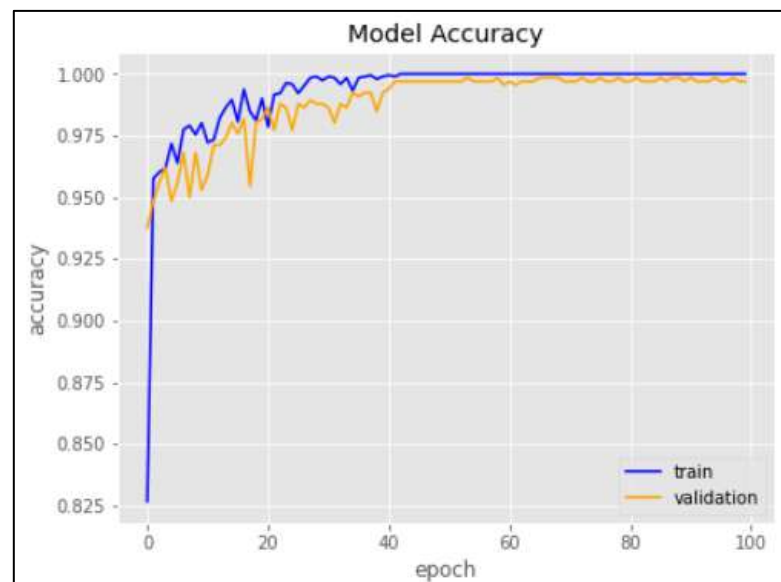


Figura 161. Gráfica de *Accuracy VGG19* (meandros) – *Dataset Perú*

Fuente: Elaboración propia

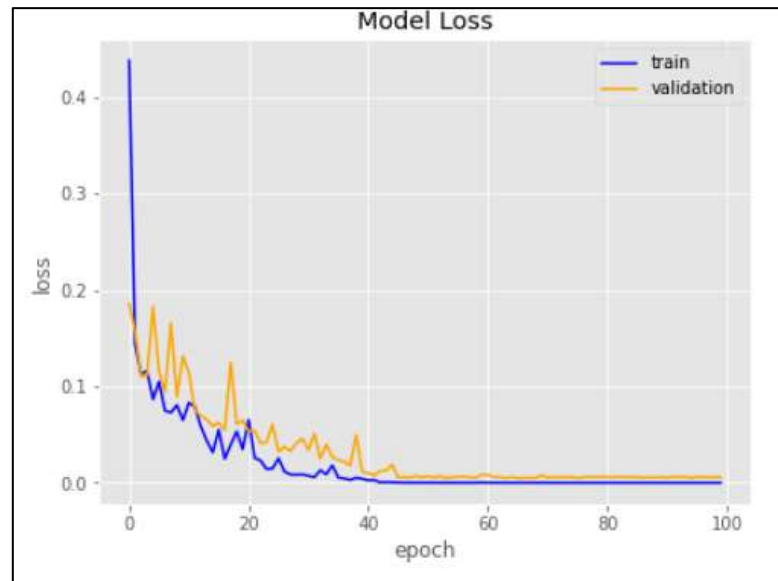


Figura 162. Gráfica de *Loss* VGG16 (meandros) – *Dataset* Perú

Fuente: Elaboración propia

Inception

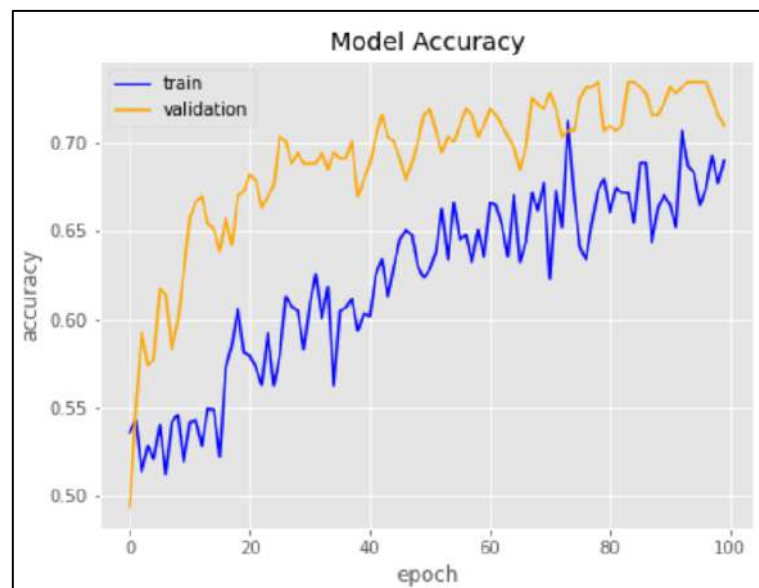


Figura 163. Gráfica de *Accuracy* Inception (meandros) – *Dataset* Perú

Fuente: Elaboración propia

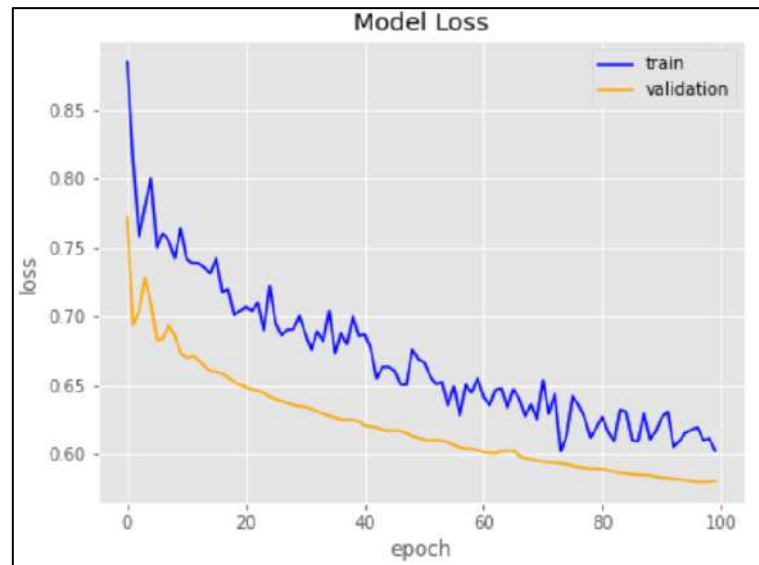


Figura 164. Gráfica de *Loss* Inception (meandros) – *Dataset* Perú

Fuente: Elaboración propia

ResNet

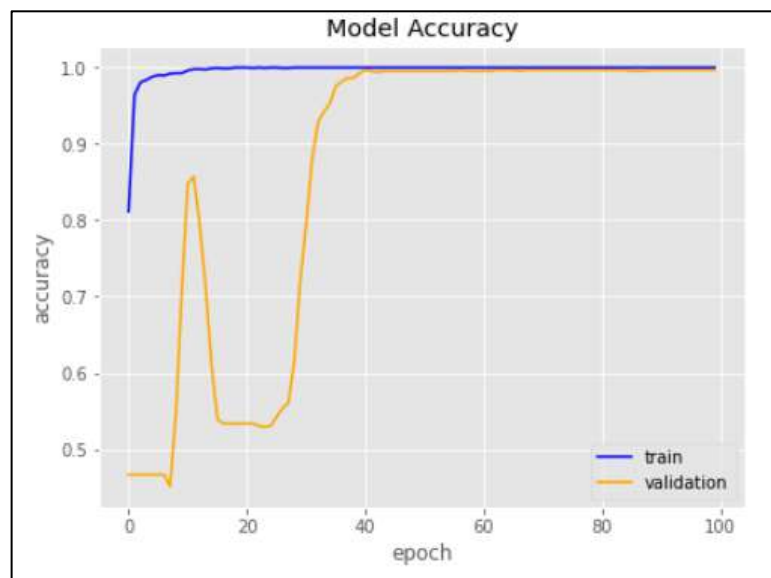


Figura 165. Gráfica de *Accuracy* ResNet (meandros) – *Dataset* Perú

Fuente: Elaboración propia

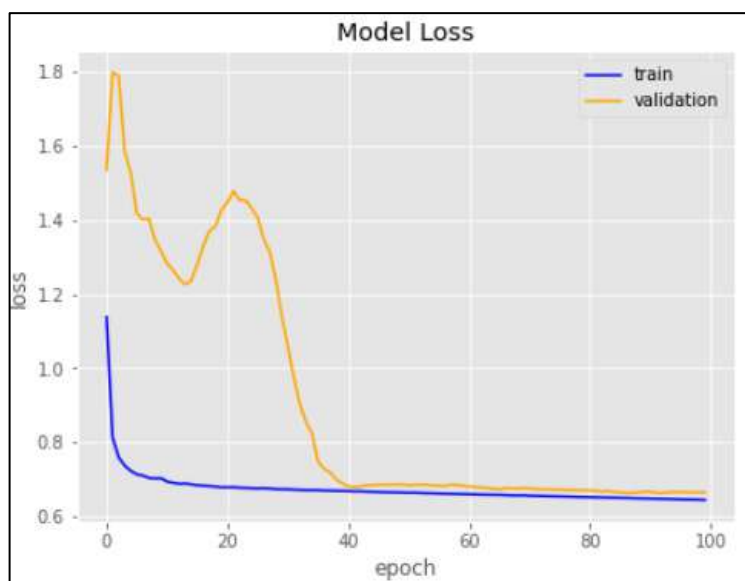


Figura 166. Gráfica de *Loss* ResNet (meandros) – *Dataset* Perú

Fuente: Elaboración propia

LeNet

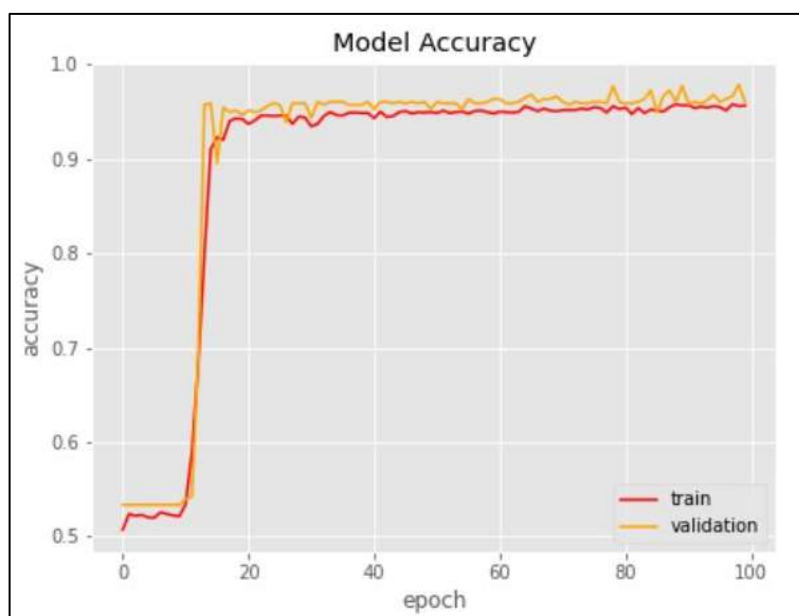


Figura 167. Gráfica de *Accuracy* LeNet (meandros) – *Dataset* Perú

Fuente: Elaboración propia

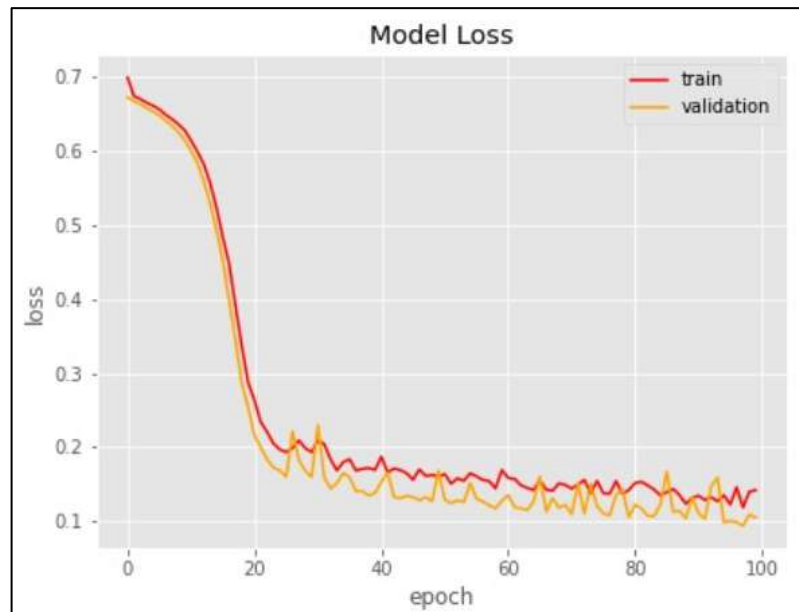


Figura 168. Gráfica de *Loss* LeNet (meandros) – *Dataset Perú*

Fuente: Elaboración propia

Por lo tanto, se puede inferir que los mejores resultados se obtuvieron de la red VGG19, con 99.7% de *Accuracy*, cuya configuración fue *learning_rate* = 0.001, *epochs*=100 y *batch_size* =64. Además, la red que obtuvo menores resultados fue ResNet50, esto debido a la gran cantidad de capas que posee la red, por lo cual el modelo presenta *overfitting* y esto también se valida observando que el los bajó hasta aproximadamente 0.6.

Espirales:

Tabla 40.

Resultados con CNN (espirales) – Dataset Perú

	Accuracy	Precision	Recall	F1-Score
VGG16	0.99	0.99	1.00	0.99
VGG19	0.99	0.98	1.00	0.99
ResNet	0.99	0.99	1.00	0.99
Inception	0.95	0.91	0.99	0.95
LeNet	0.93	0.87	0.99	0.93

Fuente: Elaboración propia

Las siguientes gráficas muestran cómo es que las redes se entrenaron y sus resultados respecto al *Accuracy* y *Loss*.

VGG16

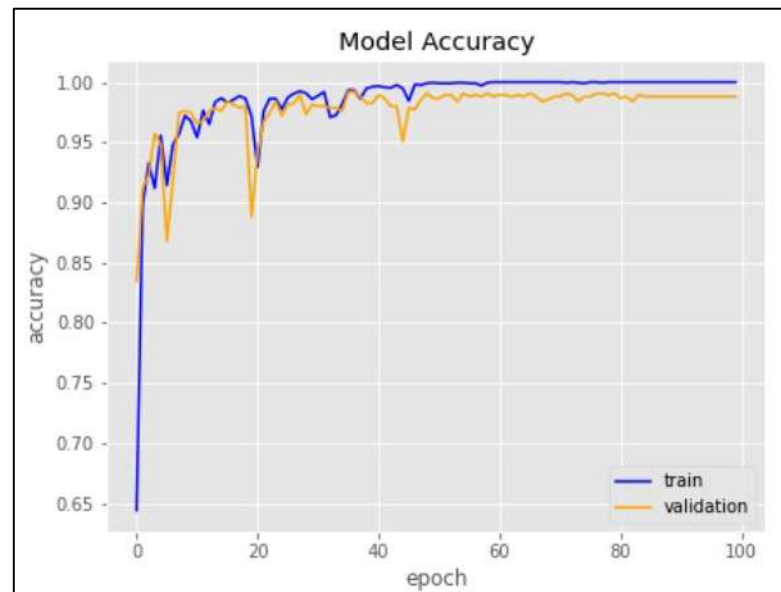


Figura 169. Gráfica de *Accuracy* VGG16 (espirales) – *Dataset* Perú

Fuente: Elaboración propia

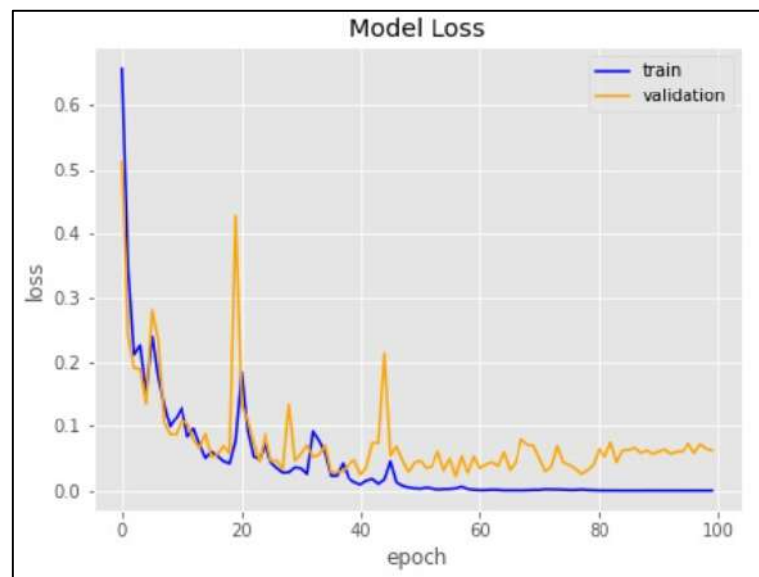


Figura 170. Gráfica de *Loss* VGG16 (espirales) – *Dataset* Perú

Fuente: Elaboración propia

VGG19

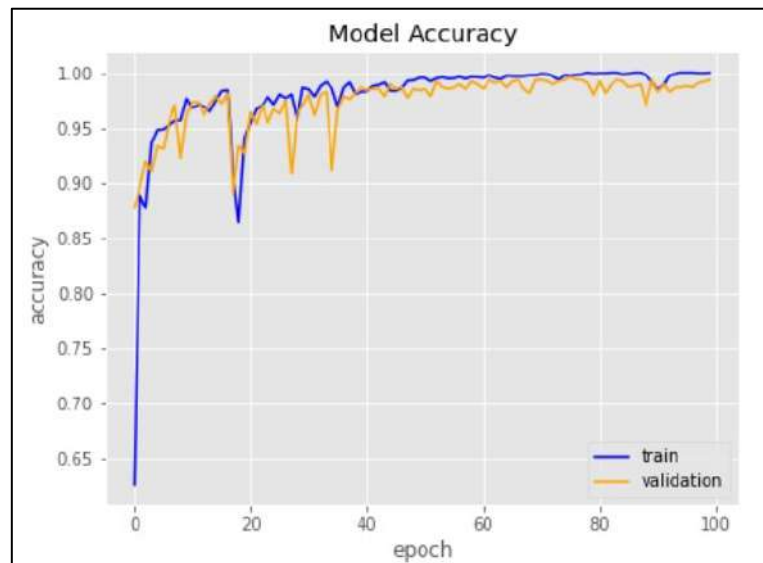


Figura 171. Gráfica de *Accuracy* VGG19 (espirales) – *Dataset Perú*

Fuente: Elaboración propia

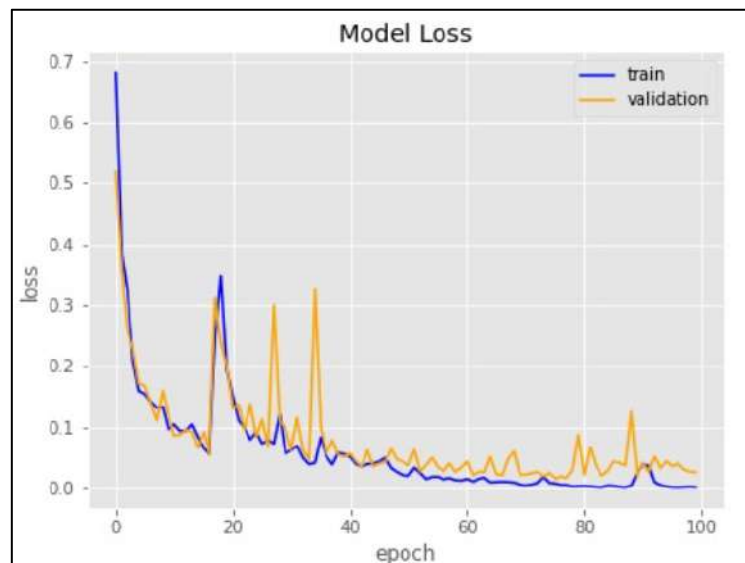


Figura 172. Gráfica de *Loss* VGG19 (espirales) – *Dataset Perú*

Fuente: Elaboración propia

Inception

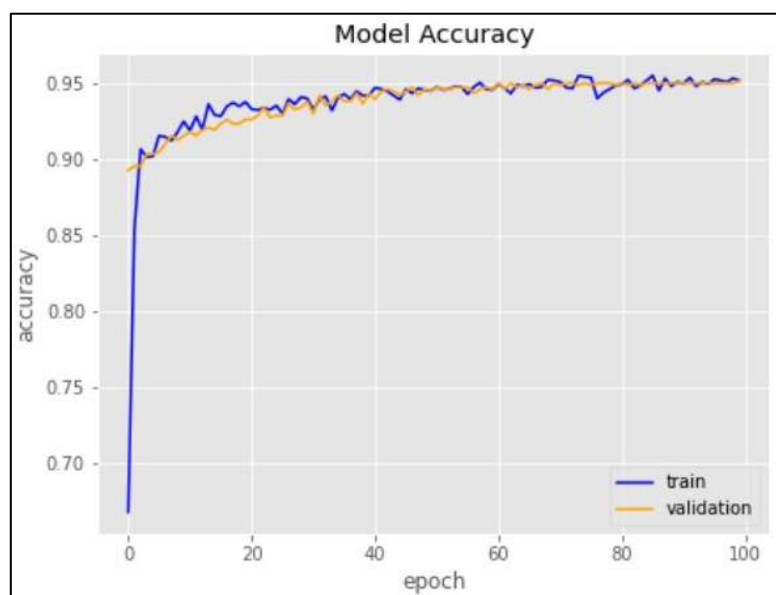


Figura 173. Gráfica de Accuracy Inception (espirales) – Dataset Perú

Fuente: Elaboración propia

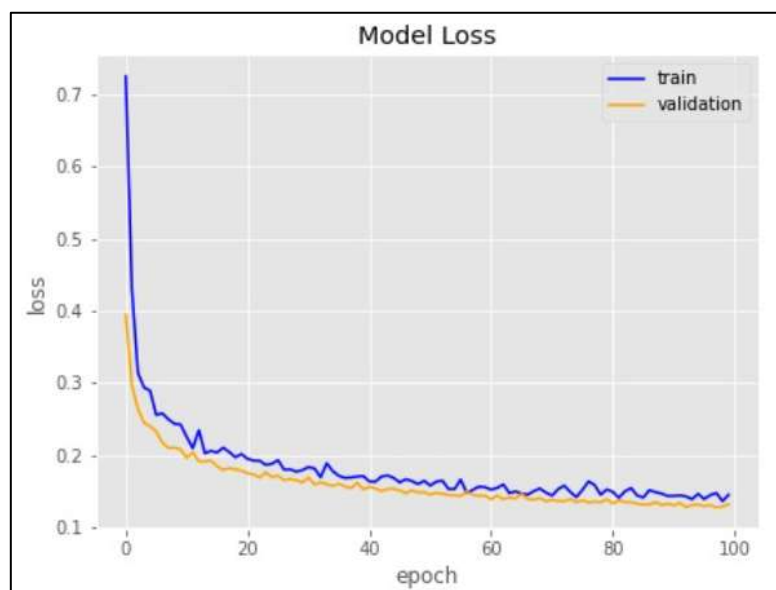


Figura 174. Gráfica de Loss Inception (espirales) – Dataset Perú

Fuente: Elaboración propia

ResNet

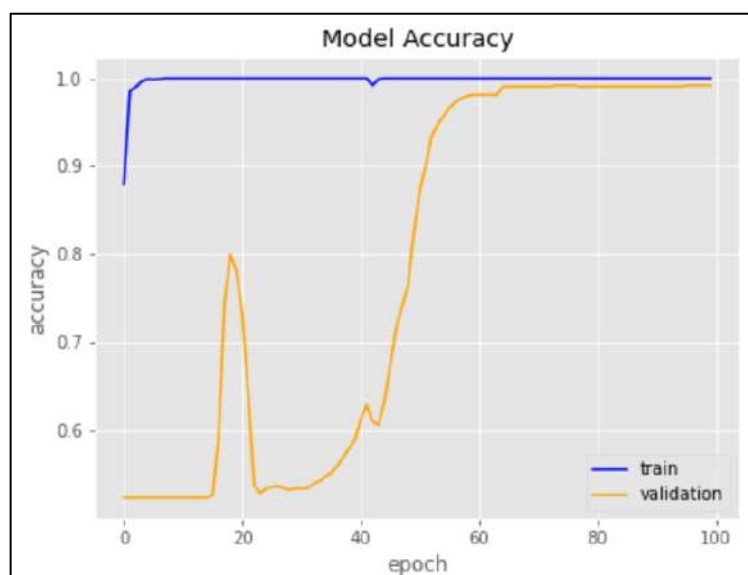


Figura 175. Gráfica de *Accuracy* ResNet (espirales) – *Dataset Perú*

Fuente: Elaboración propia

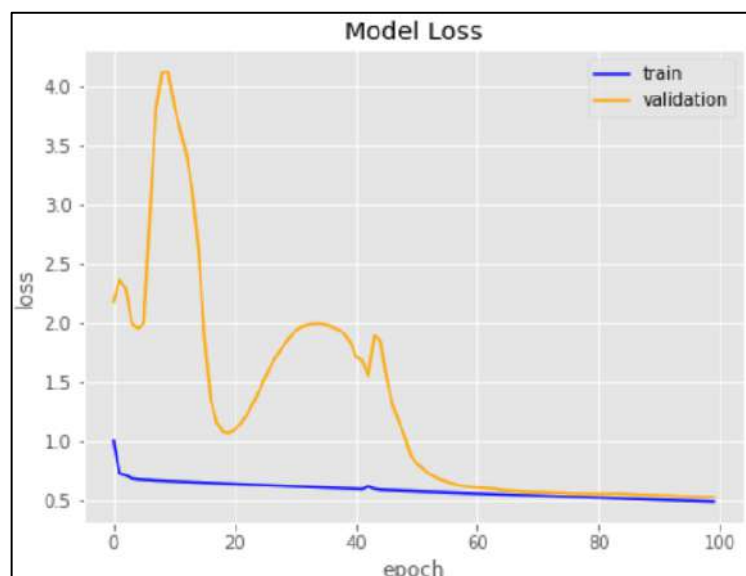


Figura 176. Gráfica de *Loss* ResNet (espirales) – *Dataset Perú*

Fuente: Elaboración propia

LeNet

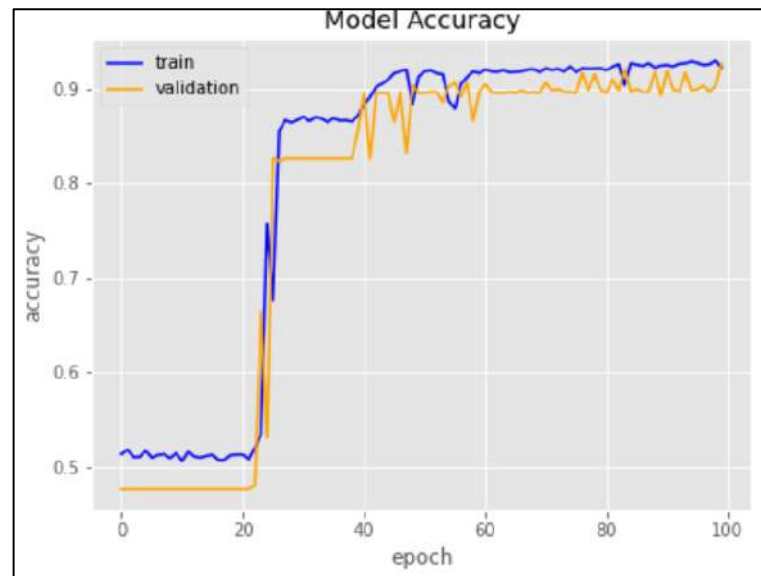


Figura 177. Gráfica de *Accuracy* LeNet (espirales) – *Dataset Perú*

Fuente: Elaboración propia

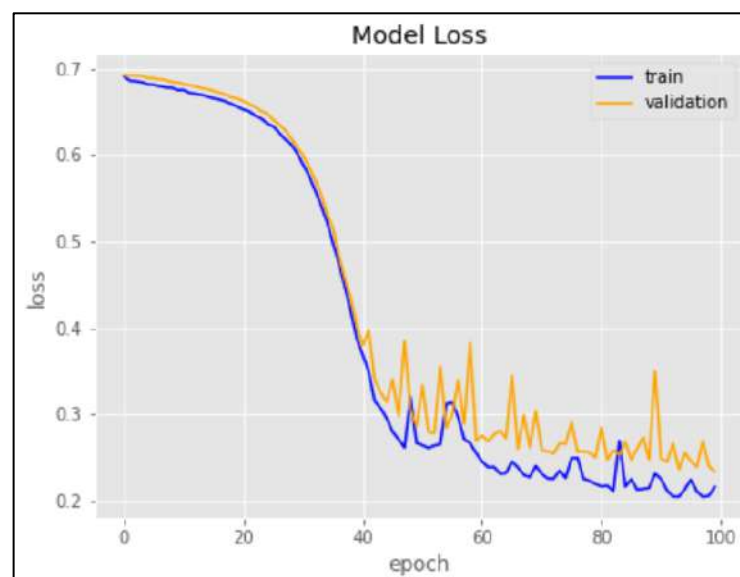


Figura 178. Gráfica de *Loss* LeNet (espirales) – *Dataset Perú*

Fuente: Elaboración propia

Por lo tanto, se puede inferir que los mejores resultados se obtuvieron de la red VGG16, con 99% de *Accuracy*, cuya configuración fue `learning_rate = 0.01`, `epochs=100` y `batch_size =128`. Además, la red que obtuvo menores resultados fue ResNet50, esto debido a la gran

cantidad de capas que posee la red, por lo cual el modelo presenta *overfitting* y esto también se valida porque se observa un los de aproximadamente 0.5.

- Metodología 3:

Respecto a las redes neuronales convolucionales con ML se obtuvieron los siguientes resultados, tanto para meandros como para espirales. La matriz de confusión para cada experimento de meandros y espirales se puede observar en el Anexo 12.

MEANDROS:

Tabla 41.

Resultados con CNN + ML (meandros) – Dataset Perú

		Accuracy	Precision	Recall	F1-Score	AUC
VGG16	SVM	0.83	0.89	0.86	0.87	0.86
	RF	1.00	1.00	1.00	1.00	0.99
	KNN	1.00	1.00	1.00	0.99	0.99
VGG19	SVM	0.86	0.90	0.88	0.89	0.88
	RF	0.83	0.89	0.85	0.87	0.86
	KNN	0.84	0.89	0.86	0.88	0.86
ResNet	SVM	0.84	0.89	0.90	0.90	0.86
	RF	0.86	0.90	0.88	0.89	0.88
	KNN	0.83	0.89	0.85	0.87	0.86
Inception	SVM	0.75	0.73	0.77	0.75	0.74
	RF	0.80	0.81	0.81	0.81	0.81
	KNN	0.78	0.77	0.79	0.78	0.77
LeNet	SVM	0.94	1.00	0.95	0.97	0.97
	RF	0.96	1.00	0.97	0.98	0.98
	KNN	0.97	1.00	0.98	0.99	0.98

Fuente: Elaboración propia

– Dataset Perú

Se infiere, que los mejores resultados para meandros han sido obtenidos de la penúltima capa de la red VGG16, con 512 columnas de vectores característicos, a los cuales se le agregó la columna target y finalmente se aplicó algoritmos de ML, de los cuales se obtuvo el mejor

resultado 99.9% de *Accuracy* con RF, cuya configuración es de *criterion='gini'*, *max_depth=4*, *max_features='auto'* y *n_estimators=200*.

ESPIRALES:

Tabla 42.

Resultados con CNN + ML (meandros) – Dataset Perú

		Accuracy	Precision	Recall	F1-Score	AUC
VGG16	SVM	0.98	0.99	0.99	0.99	0.98
	RF	0.99	1.00	0.99	0.99	0.99
	KNN	0.97	0.98	0.98	0.98	0.97
VGG19	SVM	1.00	0.99	1.00	1.00	0.99
	RF	0.99	0.99	0.99	0.99	0.99
	KNN	0.99	0.98	1.00	0.99	0.99
ResNet	SVM	1.00	0.78	1.00	0.88	0.85
	RF	1.00	0.76	1.00	0.86	0.84
	KNN	1.00	0.78	1.00	0.88	0.85
Inception	SVM	0.98	0.94	0.98	0.96	0.96
	RF	0.99	0.95	0.99	0.97	0.97
	KNN	0.98	0.94	0.98	0.96	0.96
LeNet	SVM	0.99	0.95	0.99	0.97	0.97
	RF	0.99	0.93	1.00	0.96	0.95
	KNN	0.99	0.96	1.00	0.98	0.97

Fuente: Elaboración propia

Se infiere, que los mejores resultados para meandros han sido obtenidos de la penúltima capa de la red VGG19, con 512 columnas de vectores característicos, a los cuales se le agregó la columna target y finalmente se aplicó algoritmos de ML, de los cuales se obtuvo el mejor resultado 100% de *Accuracy* con SVM, cuya configuración es de *kernel='linear'*, *C=1* y *gamma=0.1*.

5.3 Despliegue

En este apartado se observa cómo se planteó la usabilidad del modelo predictivo.

5.3.1 arquitectura para el usuario

En esta primera arquitectura se puede visualizar cómo es que un usuario puede obtener información sobre el Parkinson, hacerse el examen de escritura o buscar ayuda personalizada a través de un asistente virtual (AV) en Facebook. Asimismo, es importante detallar que cuando el usuario desee realizar un examen de escritura, para el análisis de la imagen y la predicción se consulta al modelo predictivo que obtuvo mejor performance y que se desarrolló previamente en el capítulo anterior. Además, si el usuario después de hacer el examen de escritura obtiene predicción positiva o negativa, es decir, que tiende a presentar la Enfermedad de Parkinson o no, el asistente virtual le pedirá sus datos personales y si el usuario acepta estos serán guardados en una base de datos, junto con las respuestas de los síntomas principales. Finalmente, un miembro de la Asociación podrá analizar los datos de estas personas para poder darles atención o guiarlas en qué procedimientos deberían realizar.

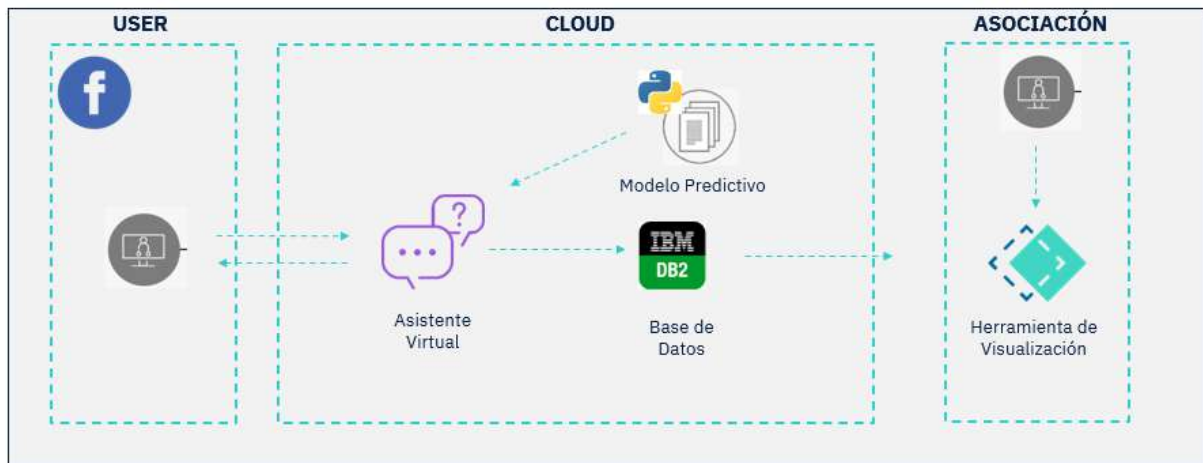


Figura 179. Arquitectura para el usuario

Fuente: Elaboración propia

- Descripción del asistente virtual:

El asistente virtual fue construido a través del servicio “*Watson Assistant*” de la *Cloud* de IBM. Cuando un usuario empieza a interactuar con el asistente virtual, lo primero que este le dice es en qué le puede ayudar y le da tres opciones para empezar a interactuar. (Figura 180). Cabe indicar que, desde la primera interacción con el AV, se guarda en una variable de contexto la fecha de ese día.

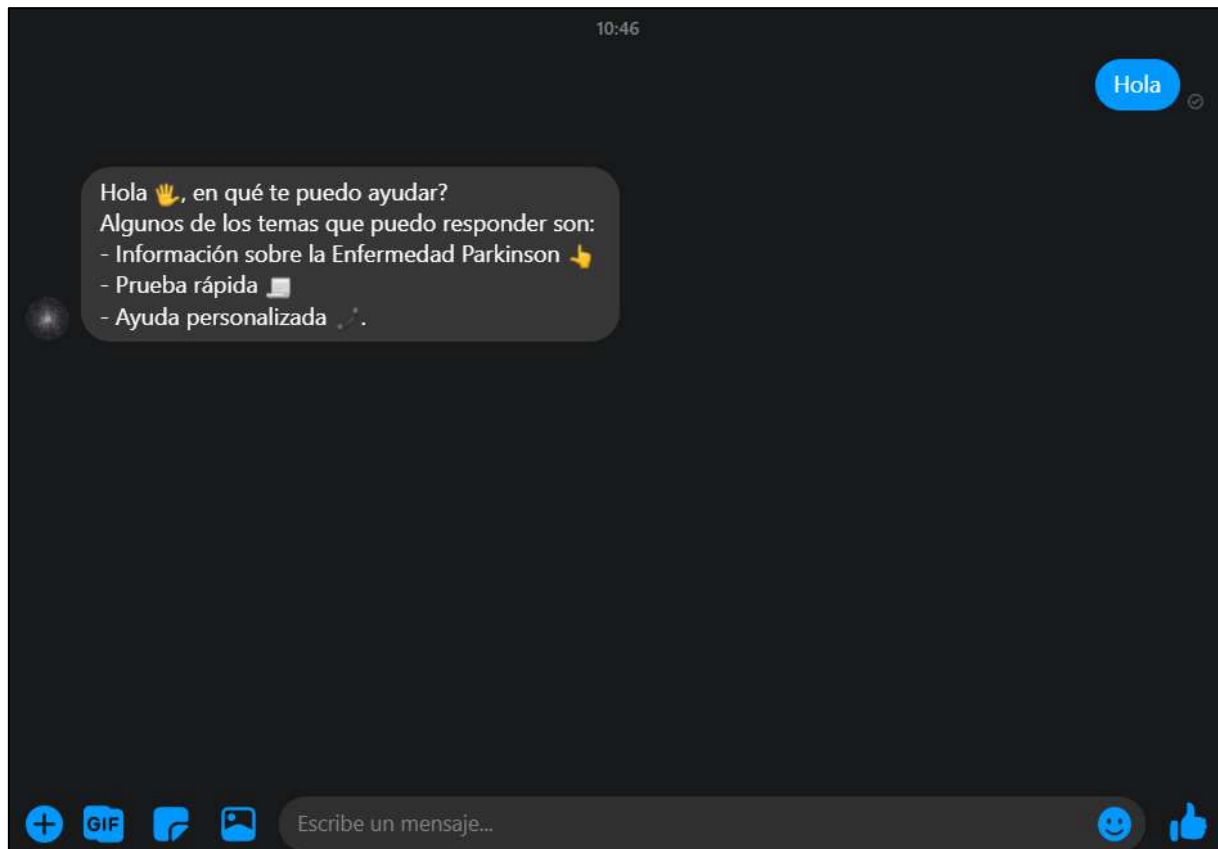


Figura 180. Primera interacción con el asistente

Fuente: Elaboración propia

Seguidamente, si el usuario elige la primera opción para saber Información sobre la Enfermedad de Parkinson, el asistente responde con información general sobre la EP y luego te vuelve a mostrar las opciones sobre cómo puede ayudar. (Figura 181)

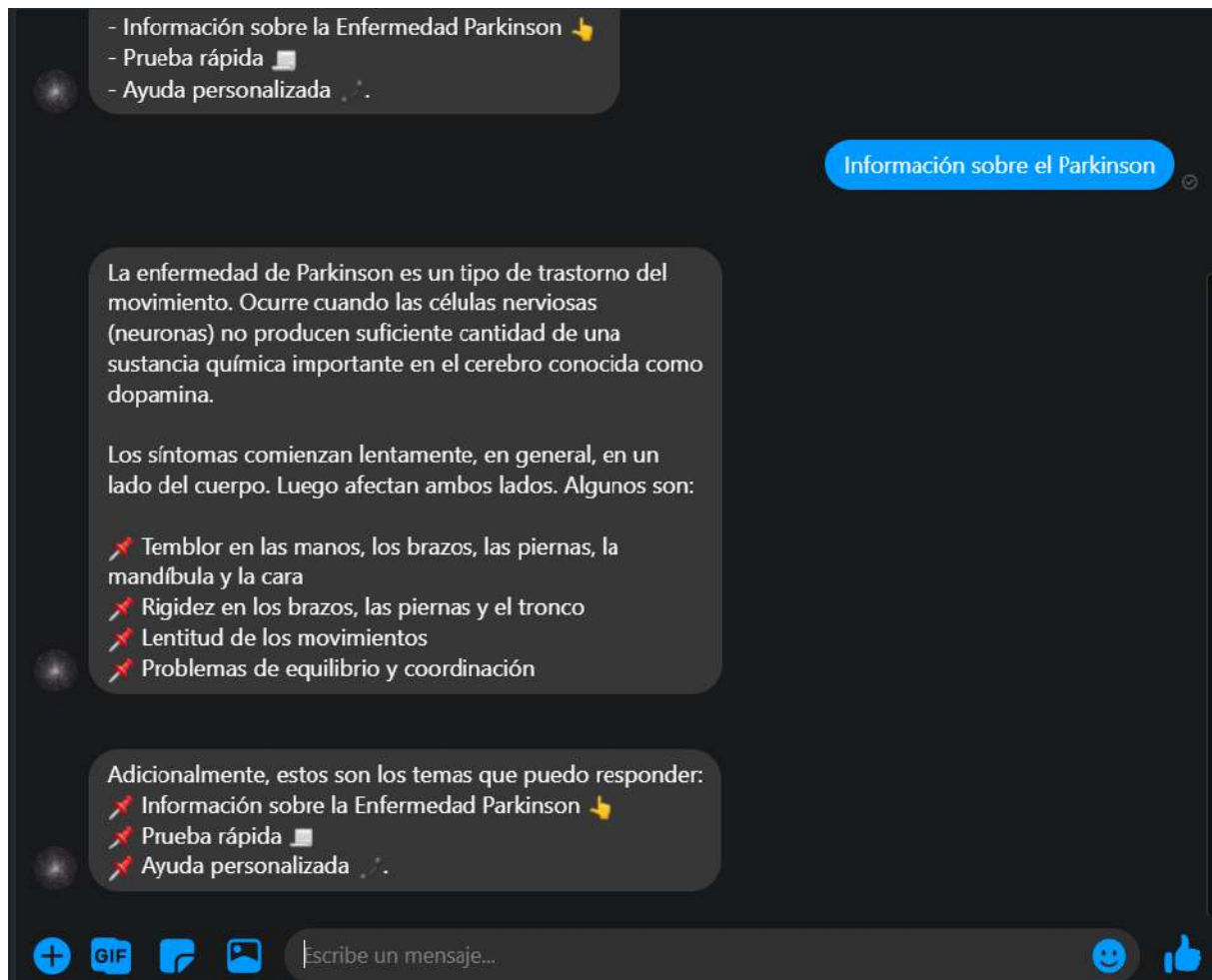


Figura 181. Opción Información sobre el Parkinson

Fuente: Elaboración propia

Si el usuario quiere hacerse una prueba escrita rápida online tiene que escribirle al asistente y este responde diciendo que va a empezar a aplicar la prueba y que responda una serie de preguntas, también le dice que va a ser completamente anónima si el usuario lo desea. Las preguntas que le hace son las siguientes:

- Últimamente, ha sentido temblor en las manos, ¿los brazos, las piernas, la mandíbula o la cara?
- ¿Ha sentido rigidez en los brazos, las piernas o el tronco?
- ¿Ha sentido lentitud de los movimientos?
- Finalmente, ¿Ha sentido problemas de equilibrio o coordinación?

Dependiendo de las respuestas del usuario, el asistente graba en variables de contexto lo que va respondiendo. Además, dependiendo de lo que responde se le aplica o no el examen escrito. En la Tabla 43 se observan las posibles combinaciones de respuesta y las variables

de contexto que se estarían guardando, las cuales son Respuesta1, Respuesta2, Respuesta3, Respuesta4, prob y parkinson.

Tabla 43.

Flujo de preguntas de síntomas de la EP

Respuesta1	Respuesta2	Respuesta3	Respuesta4	Prueba	prob	parkinson
si	No respondió	No respondió	No respondió	Sí se le hace el examen escrito	Probabilidad que arroja el modelo predictivo	“si” o “no” dependiendo del modelo predictivo
no	si	No respondió	No respondió	Sí se le hace el examen escrito	Probabilidad que arroja el modelo predictivo	“si” o “no” dependiendo del modelo predictivo
no	no	si	No respondió	Sí se le hace el examen escrito	Probabilidad que arroja el modelo predictivo	“si” o “no” dependiendo del modelo predictivo
no	no	no	no	No se le hace el examen escrito	No se le aplicó la prueba	No se le aplicó la prueba

Fuente: Elaboración propia

En el siguiente ejemplo de interacción, el usuario responde que no al primer síntoma, pero sí al segundo, es por ello que el asistente le envía el examen online. (Figura 182)

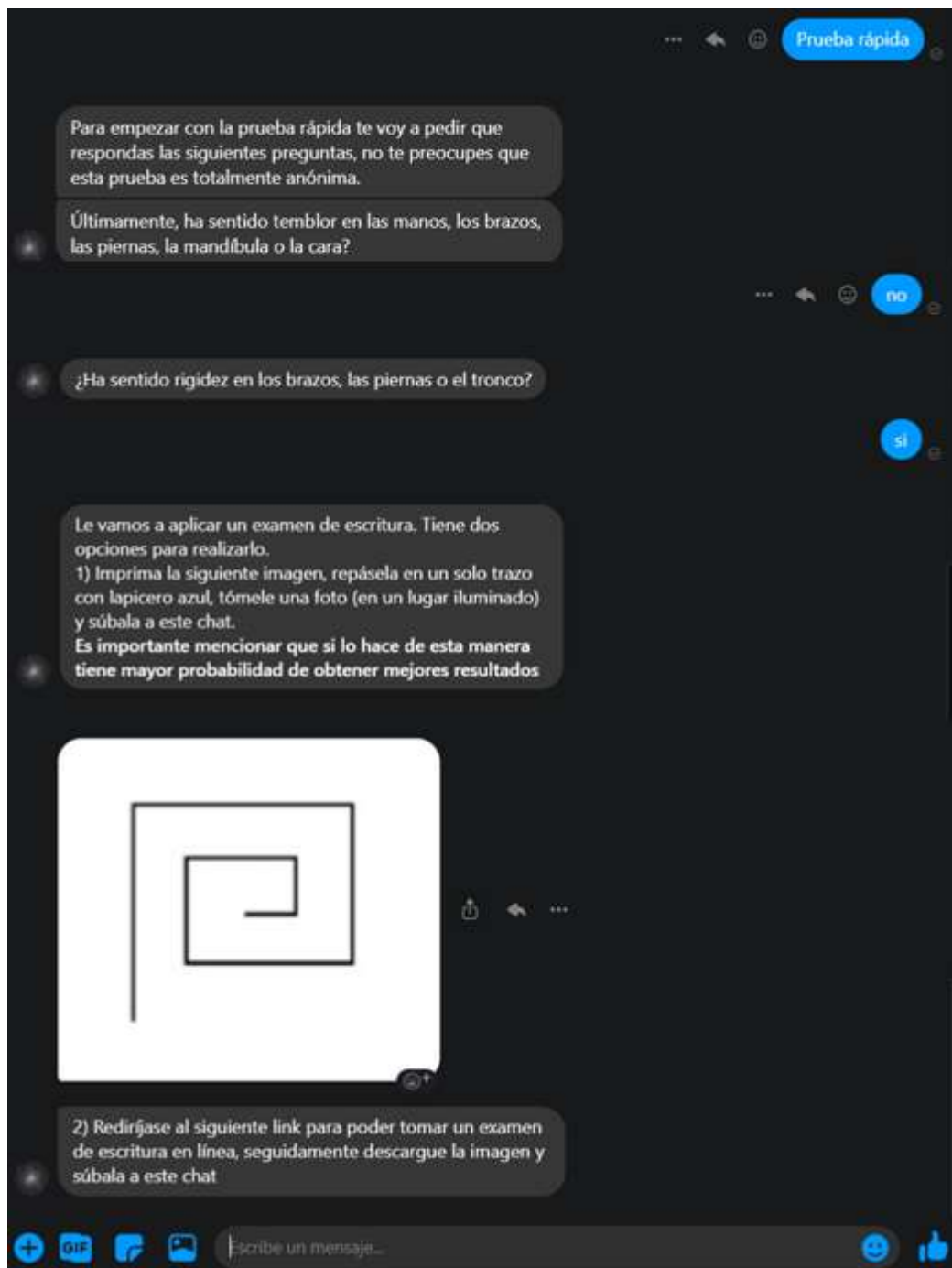


Figura 182. Opciones de cómo se debe realizar la Prueba Rápida

Fuente: Elaboración propia

En este punto, el usuario tiene dos opciones para realizarlo:

Primero: Como el asistente indica en la Figura 182, el usuario tiene que imprimir la imagen adjunta, repasarla en un solo trazo con lapicero azul y finalmente, tomarle foto al examen y subirlo al chat, esta es la forma más segura para obtener un buen resultado, dado que el trazo lo estaría realizando el usuario de una forma más cómoda.

Segunda: En este caso el asistente indica que se tiene que redirigir a un enlace, el cual lleva al usuario a una interfaz en la cual encuentra la misma imagen del meandro. Si el usuario está desde celular o computadora, lo que tiene que hacer es realizar el trazo con el dedo, si es en el celular, o con el mouse, si es en la computadora. Luego, descargar la imagen y subirla al chat con el AV. En la Figura 183 se observa cómo es la interfaz y cómo el usuario ha hecho una prueba con el mouse, cabe indicar que cuando esté realizando el trazo puede darle a la opción de “Borrar”, para iniciar el trazo desde cero y puede darle a la opción de “Descargar” para descargar la imagen como png.

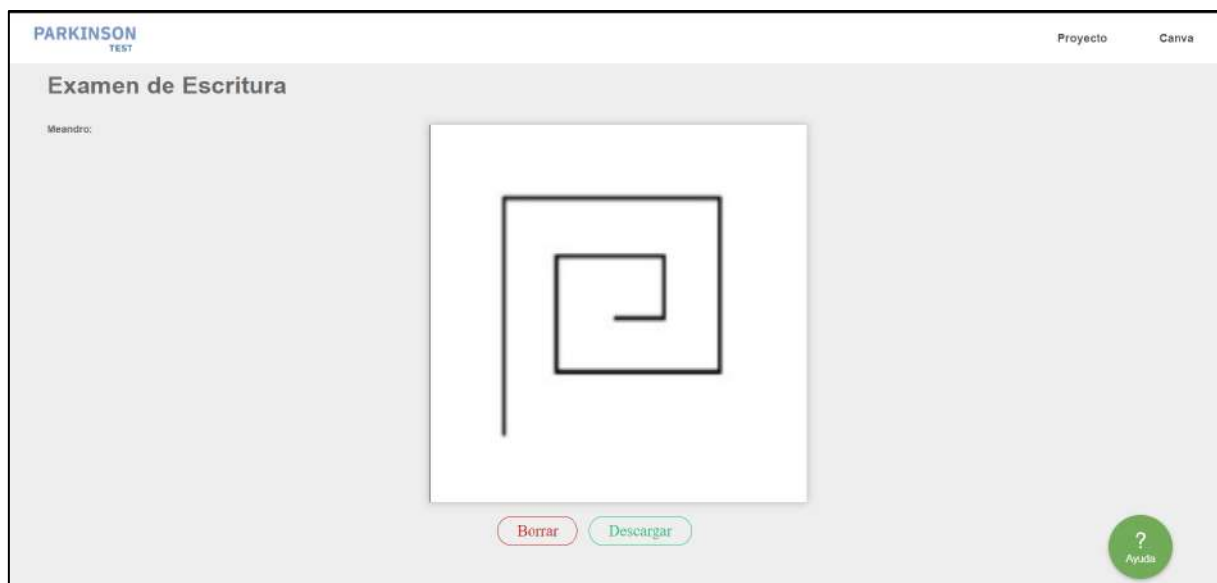


Figura 183. Examen escrito online

Fuente: Elaboración propia

Una vez que el usuario haya subido la imagen del examen con el trazo, el AV por detrás está conectado al modelo de clasificación de *Machine Learning*, el cuál fue desplegado a través de un API para poder ser expuesto como *Web Service*. Es por ello que, después que el modelo de ML preprocese la imagen y realice la predicción, devuelve el resultado de si el usuario tiende o no a presentar la EP. De ser positivo, le da una respuesta con la probabilidad a presentar esta enfermedad y le indica lo siguiente: “Usted tiende a presentar la Enfermedad de Parkinson, para

cerciorarnos queremos tener una entrevista personal o por llamada con usted. Si está de acuerdo responda sí de lo contrario no”. De ser negativo, le indica lo siguiente:” Me alegra decirle que usted no tiende a presentar síntomas de la Enfermedad de Parkinson. De todos modos, si desea puede dejarnos sus datos para seguir en contacto, de ser así responda sí, de lo contrario no”. En ambos casos el usuario es libre de responder sí o no, si responde que sí, el AV empieza a pedirle sus datos, de lo contrario se despide y agradece al usuario. En la Figura 184 se puede observar un ejemplo en la cual el usuario sube la imagen del examen escrito que realizó.

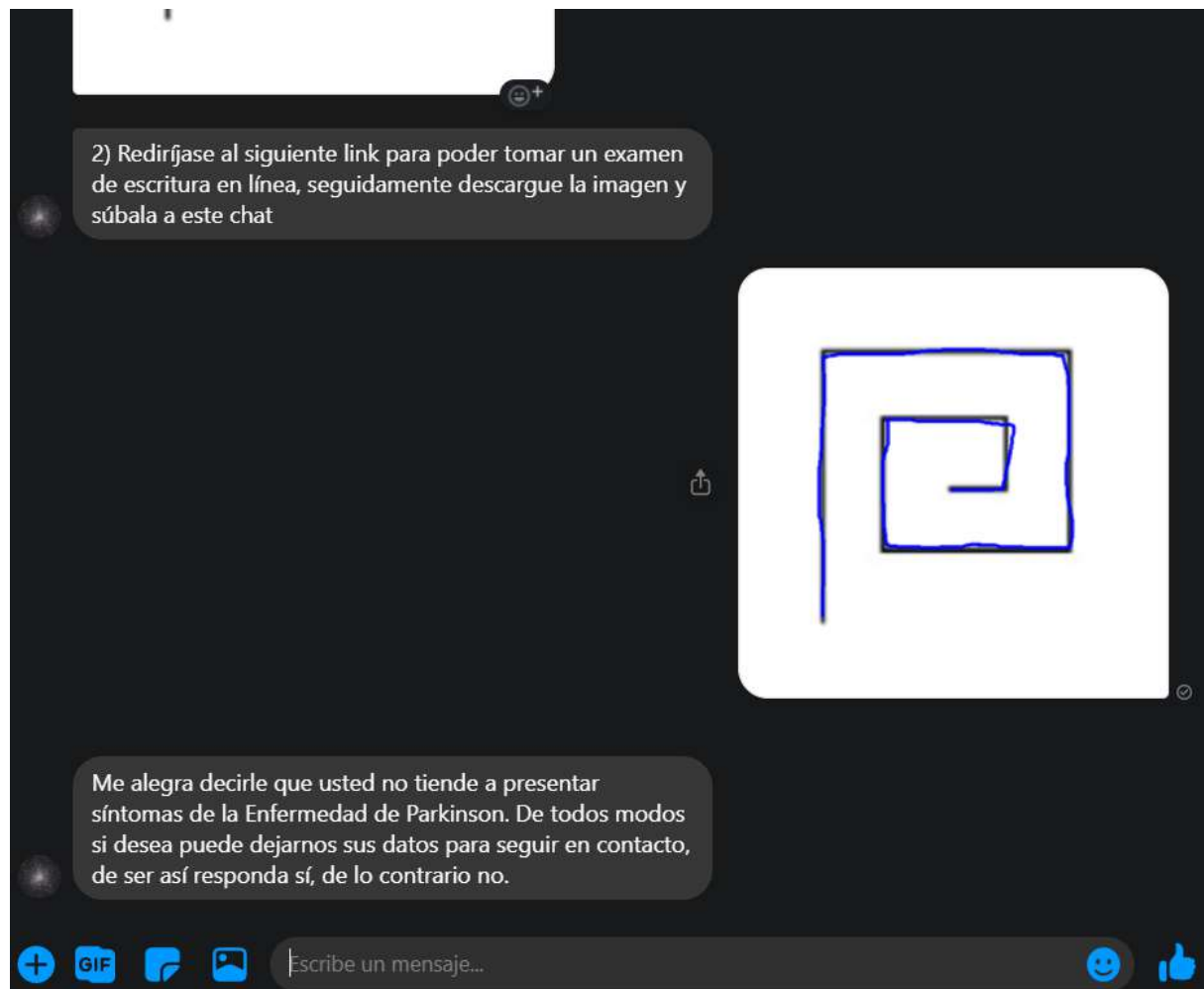


Figura 184. Ejemplo de resultado sobre la Prueba en Línea – Modelo de Clasificación

Fuente: Elaboración propia

En la Figura 185 se observa las preguntas que el AV le hace al usuario para poder guardar su información, este paso es bastante importante, ya que las mismas preguntas se le harían de haber sido su prueba positiva. Guardar estos datos va a permitir que se pueda dar una respuesta proactiva de parte de la Asociación, hacia el usuario, ya que, al dejar su teléfono y nombres, sería posible contactarlos para poder ayudarlos y validar con un examen totalmente

escrito los resultados que arrojó el AV. Las respuestas que va dando el usuario, se van guardando en variables de contexto, las cuales son nombres, dni, teléfono, edad, ciudad.

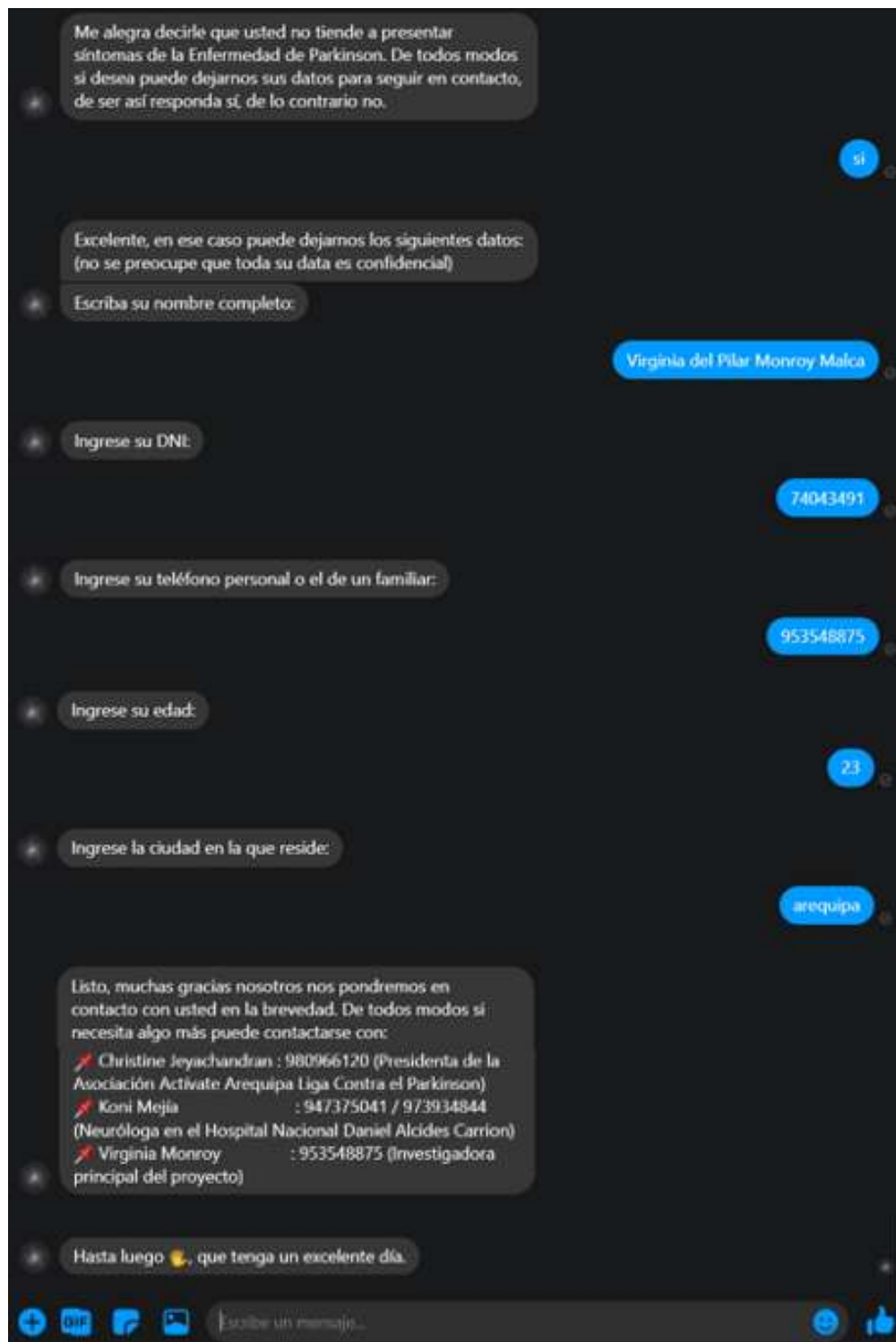


Figura 185. Recolección de datos personales a través del AV

Fuente: Elaboración propia

Si el usuario hubiese preguntado al AV por ayuda personalizada, este le hubiese respondido con los datos de tres personas a las cuales puede contactar (Figura 186). En los Anexos 9 y 10 se puede observar la autorización para poder publicar sus datos.

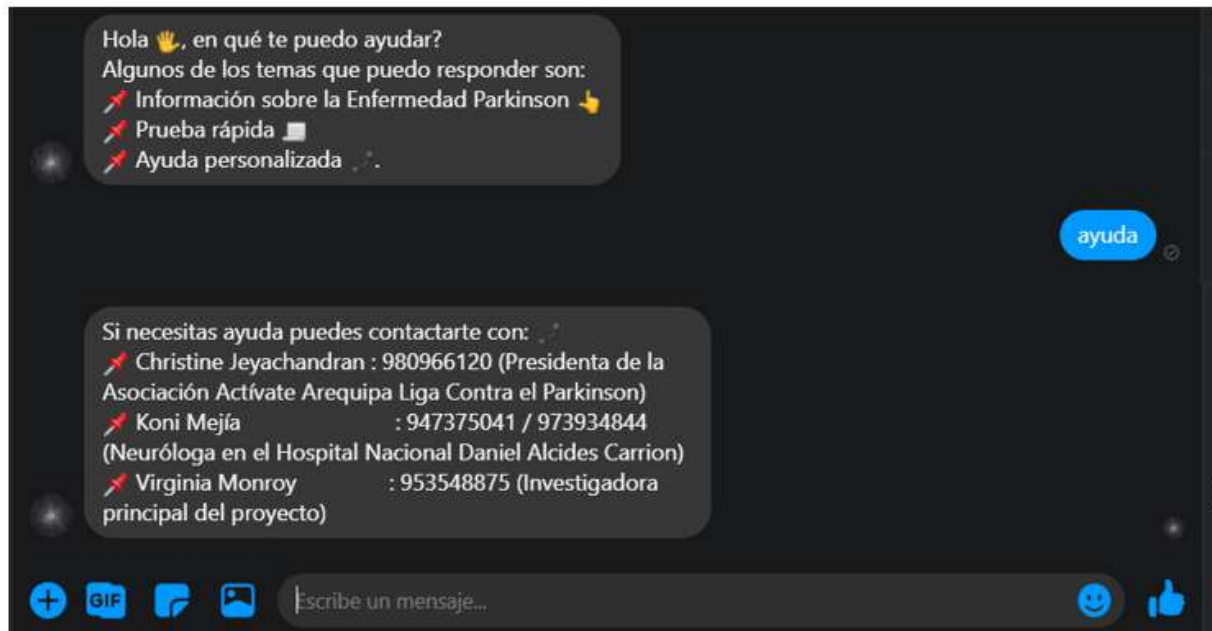


Figura 186. Opción Ayuda Personalizada

Fuente: Elaboración propia

Como se mencionó dentro del AV, existen variables de contexto que han ido guardando las respuestas y datos personales del usuario, siempre y cuando este dio la autorización, en memoria. En el momento que el usuario responda la última pregunta de datos personales, “Ingresa la ciudad en la que reside” automáticamente se hace un llamado a la base de datos DB2 y se ingresa una sentencia de *INSERT* (Figura 187).

```
1 var aux = msg.payload.context.aux
2
3 if(aux === 1){
4     var nombres = msg.payload.context.nombres
5     var fecha = msg.payload.context.fecha
6     var dni = msg.payload.context.dni
7     var telefono = msg.payload.context.telefono
8     var edad = msg.payload.context.edad
9     var ciudad = msg.payload.context.ciudad
10    var Respuesta1 = msg.payload.context.Respuesta1
11    var Respuesta2 = msg.payload.context.Respuesta2
12    var Respuesta3 = msg.payload.context.Respuesta3
13    var Respuesta4 = msg.payload.context.Respuesta4
14    var prob = msg.payload.context.prob
15    var parkinson = msg.payload.context.parkinson
16    msg.payload = "insert into CZP57479.EPA values ('"+nombres+"',
17    '"+fecha+"', '"+dni+"', '"+telefono+"', '"+edad+"',
18    '"+ciudad+"', '"+Respuesta1+"', '"+Respuesta2+"',
19    '"+Respuesta3+"', '"+Respuesta4+"', '"+prob+"',
20    '"+parkinson+"')";
21 }
22 return msg;
```

Figura 187. Código para guardar en la base de datos

Fuente: Elaboración propia

En esta base de datos, se tiene el SCHEMA “CZP57479” y la tabla “EPA” cuya estructura se puede observar en la Figura 188:

Table Definition

EPA

Approximate 31 rows (32 KB)
Updated on 2020-11-22 03:29:58

COLUMN NAME	DATA TYPE	NULLABLE	LENGTH	SCALE
NOMBRES	VARCHAR	Y	100	0
FECHA	DATE	Y	4	0
DNI	VARCHAR	Y	100	0
TELEFONO	VARCHAR	Y	100	0
EDAD	VARCHAR	Y	100	0
CIUDAD	VARCHAR	Y	100	0
RESPUESTA1	VARCHAR	Y	100	0
RESPUESTA2	VARCHAR	Y	100	0
RESPUESTA3	VARCHAR	Y	100	0
RESPUESTA4	VARCHAR	Y	100	0
PROB	VARCHAR	Y	100	0
PARKINSON	VARCHAR	Y	100	0

View Data

Figura 188. Estructura tabla EPA base de datos

Fuente: Elaboración propia

En la Figura 189 se puede observar el registro que se insertó de la interacción que se tuvo con el AV anteriormente. Es importante resaltar, que los datos de las otras personas que ya han respondido están pixelados debido a que son datos sensibles.

IBM Db2 on Cloud Storage: 23%

Preferencias sobre cookies Discover

TABLES

Back

CZPS7479.EPA

NOMBRES	FECHA	DNI	TELEFONO	EDAD	CIUDAD	RESPUESTA1	RESPUESTA2	RESPUESTA3	RESPUESTA4	PROB	PARKINSON
Virginia del Pilar ...	2020-11-22	74043493	93548878	22	Periáñez	no	si	No respondió	No respondió	93	no

Figura 189. Ejemplo de registro guardado

Fuente: Elaboración propia

Hasta este punto, el despliegue del AV en Facebook, el llamado al modelo de predicción, la conexión y guardado en la base de datos DB2 han sido orquestadas por el servicio llamado Node-RED, el cual es una herramienta de programación visual que facilita la conexión de distintos servicios. En la Figura 190 muestra un diagrama de flujo donde se explica el funcionamiento del orquestador.

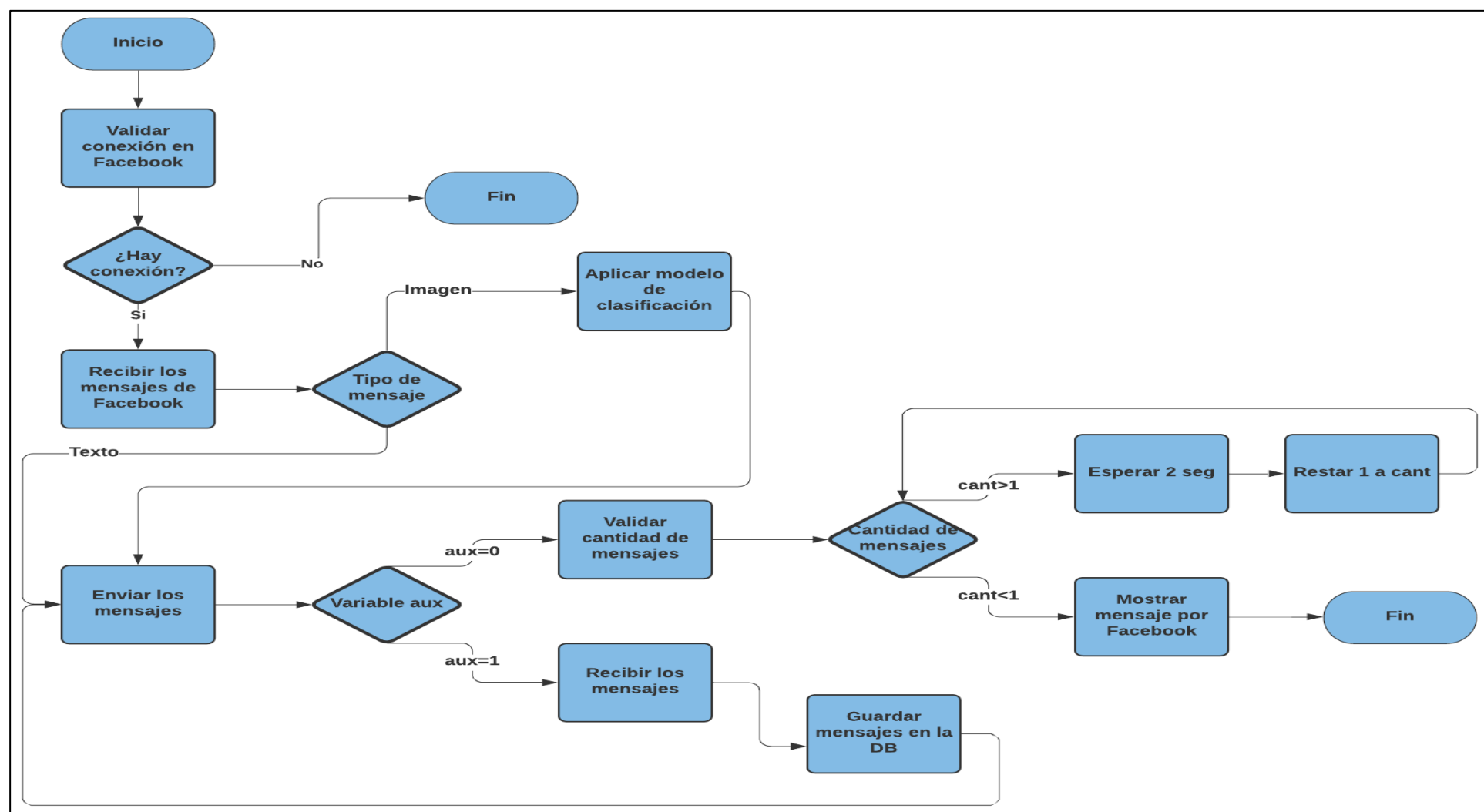


Figura 190. Diagrama de flujo del proceso en NodeRed

Fuente: Elaboración propia

En la Figura 191 se puede observar el diseño del orquestador Node-Red, la etiqueta “Verificación FB”, indica el flujo para validar la conexión entre el AV y el chat de Facebook, Messenger. Seguidamente, “Facebook” sirve para recibir los mensajes del usuario e interpretarlos, ya que estos mensajes pueden ser en formato texto o en formato imagen y la función “write text FB” sirve para imprimir los mensajes del AV en la interfaz de Facebook Messenger. Asimismo, cuando se entra al flujo de “detección imagen” es porque se ha detectado que el mensaje del usuario ha sido una imagen, por lo cual, en ese momento llama al `model_id` del modelo de predicción y devuelve un resultado de probabilidad de presentar o no la EP. La tercera etiqueta “message WA”, se activa cuando los mensajes del usuario llaman por API al AV consumiendo el servicio Watson Assistant, aquí existe una función switch, la cual tiene dos caminos: la variable de contexto `aux` tiene dos valores según avanza el flujo si es igual a “0” el flujo avanza de manera normal y se va al nodo “write text FB” para imprimir el mensaje en Facebook Messenger, si `aux` es igual a “1” significa que el usuario dio autorización para que sus respuestas y datos personales se guarden en la base de datos (Figura 185), se hace un llamado a la base de datos se guardan esos valores y luego se dirige al nodo “write text FB”. Finalmente, es importante resaltar que la función “delay 2s” sirve cuando el AV responde varios mensajes como se observa en la Figura 182, en este caso el AV respondió con un texto, imagen, texto, si no estuviera la función de delay, al tener un texto menos peso en comparación de una imagen, el AV hubiese respondido con texto, texto, imagen.

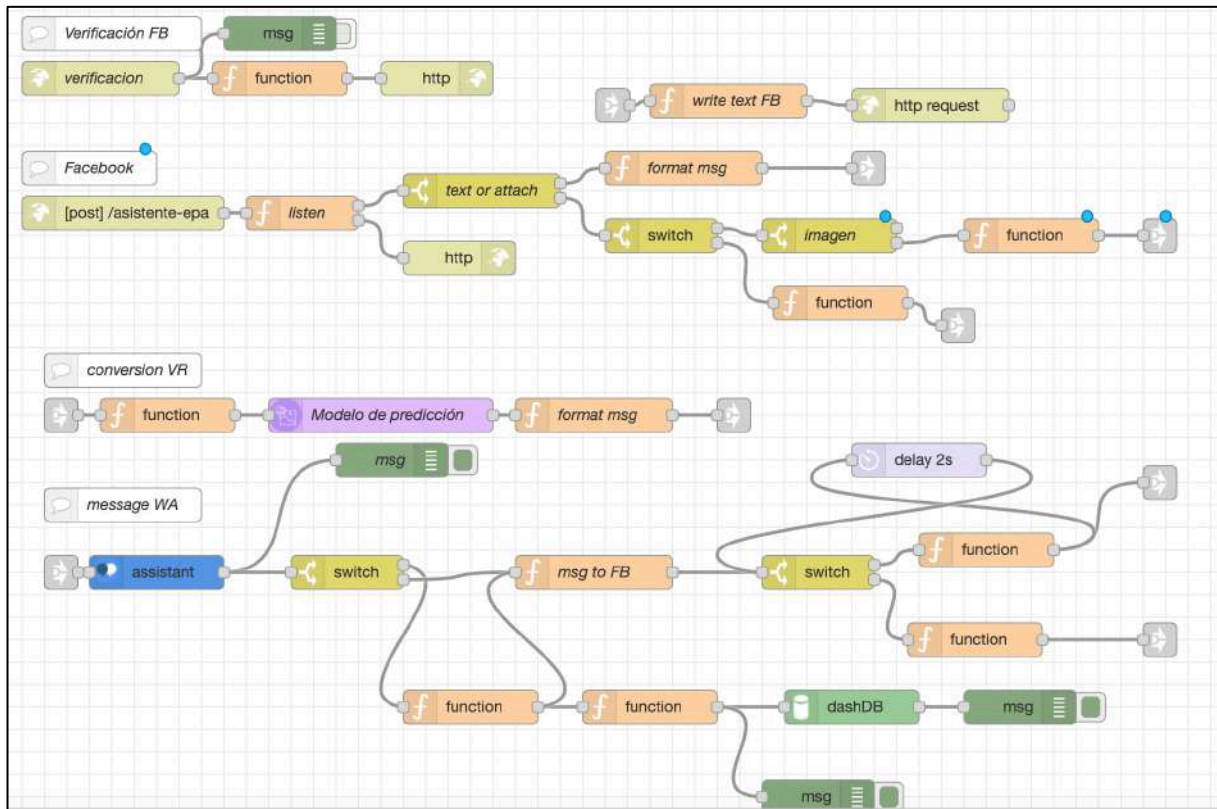


Figura 191. Arquitectura Node-RED

Fuente: Elaboración propia

Finalmente, para cerrar la Arquitectura para el usuario (Figura 179) se detalla el consumo final de los resultados de las interacciones entre el AV y los usuarios. Se hace uso del servicio *IBM Cognos Dashboard Embedded* para que la persona encargada de gestionar y monitorear a los mencionados usuarios pueda tener una herramienta “*user friendly*”. En la Figura 188 se observa el primer tab llamado “General” cuyo título es Control de Respuestas, este tab está orientado a dar datos estadísticos de todas las interacciones que se van registrando. El primer gráfico “Distribución de Respuestas” indica la cantidad de respuestas si han sido “si”, “no” o “No se le aplicó la prueba” que indican cuantas personas tienden o no a presentar la EP. Los números que se ven a la derecha indican: Cantidad de conversaciones hasta la fecha, Cantidad de personas que han interactuado con el AV a la fecha, Cantidad de Positivos, es decir que han salido positivos a la prueba y el Promedio de Edad entre las personas que han probado la herramienta. El segundo gráfico, “Cantidad de respuestas por ciudad”, resumen en un gráfico de barras la cantidad de conversaciones que se ha tenido por ciudad, hasta la fecha, la mayoría de las personas han interactuado desde Arequipa. Por último, el tercer gráfico “Cantidad de respuestas a través del tiempo según prueba” muestra cómo es la distribución de respuestas según prueba a través del tiempo.



Figura 192. Tab General del Dashboard

Fuente: Elaboración propia

En la Figura 193 se observa el segundo tab llamado “Detalle Respuestas” cuyo título es Detalle de Respuestas, este tab está orientado a mostrar en una tabla los datos y respuestas que fueron recopilados por el AV. Además, te da el link hacia lo que es la segunda arquitectura, la cual se detalla más adelante, que permite hacer una prueba en vivo de espirales y meandros. Por último, es importante resaltar que en esta tab se observan filtros los cuáles pueden ser por EDAD, CIUDAD, PARKINSON y PROB.



Figura 193. Tab Detalle Respuestas del Dashboard

Fuente: Elaboración propia

En la Figura 194 se observa el tercer tab llamado “Información” cuyo título es Información Adicional, este tab está orientado a mostrar información adicional como el detalle de las preguntas que han respondido los usuarios en la interacción con el AV, y también se observa un gráfico que muestra la distribución de las respuestas por edad.

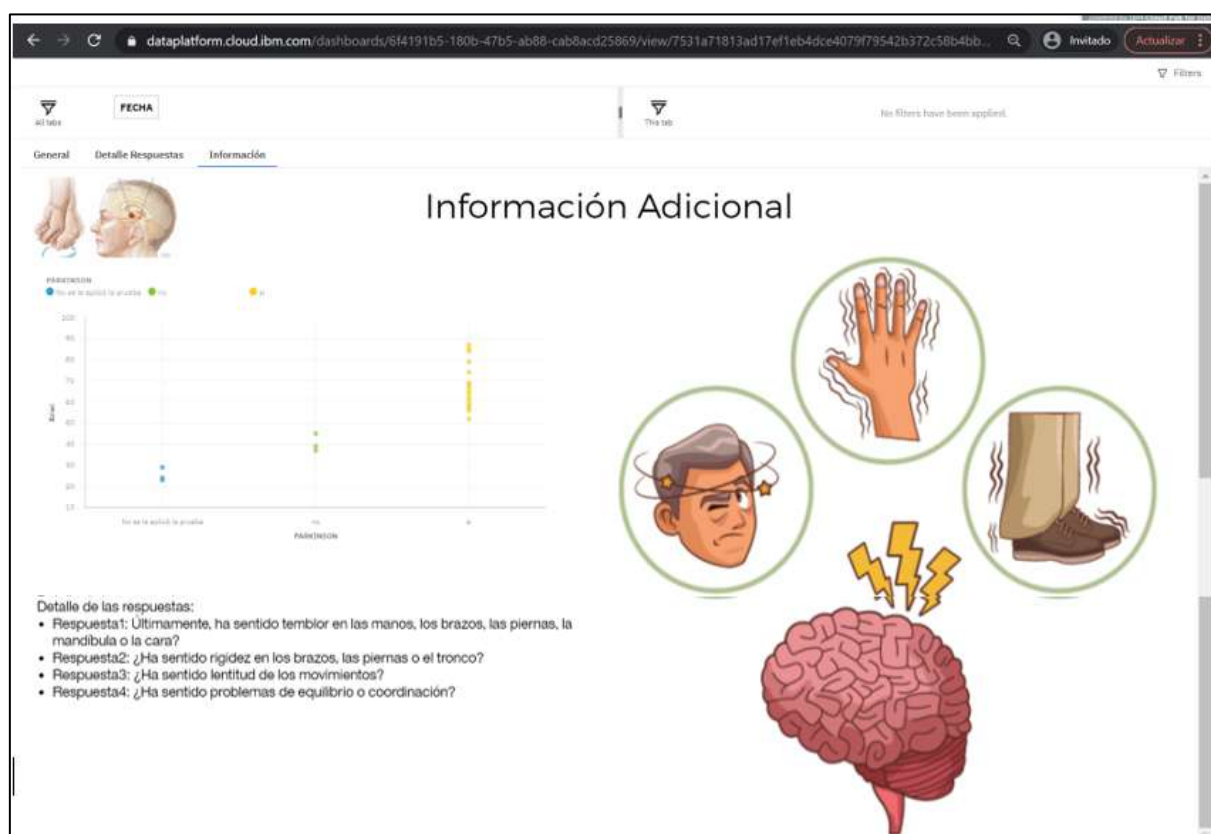


Figura 194. Tab Información Respuestas del Dashboard

Fuente: Elaboración propia

Finalmente, es importante resaltar que para los tres tabs existe el filtro de FECHA, con el cual se puede segmentar la visualización de los dashboards según fecha, esta es una opción valiosa, ya que los datos siempre cambian a través del tiempo. Asimismo, todos los gráficos son *responsive*, es decir si se ve desde el celular se adapta a la pantalla y si se da clic en algún gráfico los demás se actualizan en base a lo que se seleccionó.

5.3.2 arquitectura para el miembro de la asociación

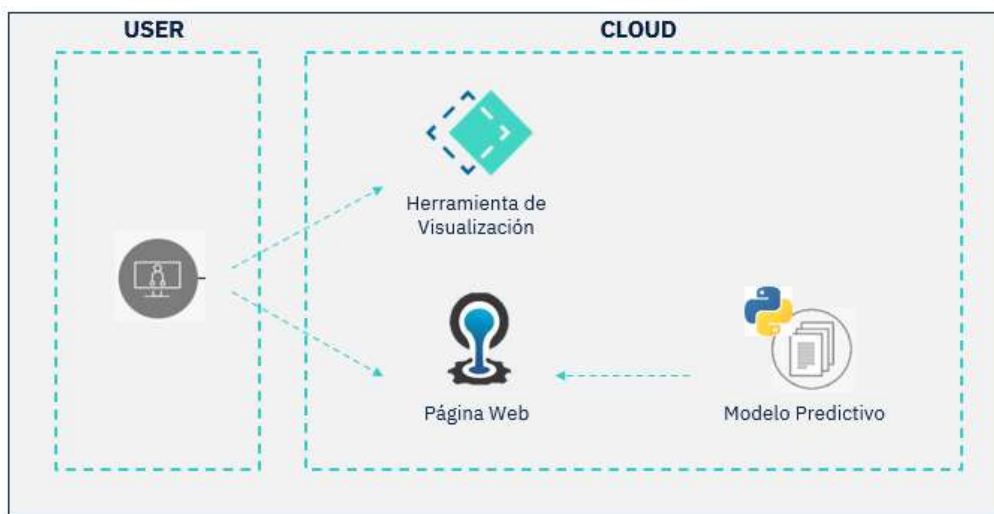


Figura 195. Arquitectura para el miembro de la Asociación

Fuente: Elaboración propia

En esta segunda arquitectura (Figura 195) se puede visualizar cómo la persona encargada de gestionar las interacciones de los usuarios con el AV puede realizar una acción proactiva o a su vez también es una herramienta reactiva. Es proactiva, ya que la persona al visualizar los gráficos y tablas en *Cognos Embedded*, puede identificar quienes son las personas que según el examen en línea han salido positivas a tender a presentar la EP y al dejar sus datos se puede contactar con ellos y citarlos. Asimismo, es reactiva, ya que los usuarios después de interactuar con el AV obtienen los datos de Christine, presidenta de la Asociación, Koni y Claudia neurólogas de profesión y los míos, principal investigadora y desarrolladora de la solución como se observa en las Figuras 185 y 186, es así que los usuarios se pueden contactar a través de una llamada y poder agendar una cita para poder tener un nuevo examen de validación y mayor información.

En ambos casos, ya sea una acción proactiva o reactiva, lo usuarios llegan con el especialista y este puede aplicarles la prueba en vivo a través de un examen de escritura como en la Figura 192 y validando que sus trazos se realicen manualmente. Luego subirlas a una página web y se llama al algoritmo de predicción por detrás y este algoritmo arroja un resultado como se observa en la Figura 193. Cabe mencionar que la página web ha sido desplegada en la plataforma como servicio código abierto *Cloud Foundry*.

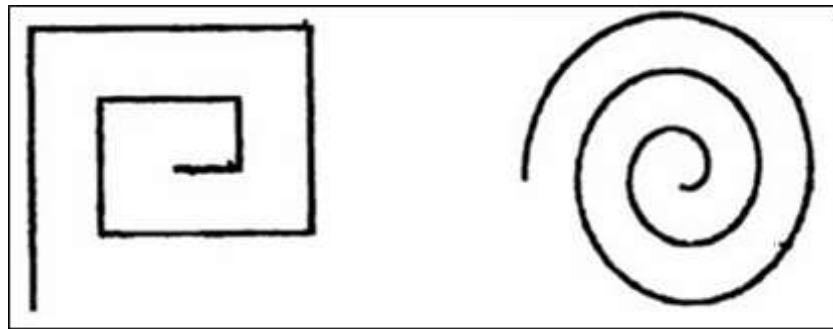


Figura 196. Examen de escritura en vivo

Fuente: Elaboración propia



Figura 197. Plataforma para el analizar las imágenes y realizar la clasificación

Fuente: Elaboración propia

Por último, esta arquitectura tiene el objetivo de revalidar los resultados que se obtuvieron en la interacción con el AV, ya que este va a ser en presencia de una especialista en neurología. Asimismo, el implementar el asistente virtual ha sido para poder facilitar la llegada a los usuarios del examen en línea, ya que en estos tiempos de pandemia o incluso cuando todo era normal las personas tienden a resistirse de ir a un doctor cuando sientan pequeñas molestias, y cuando deciden hacerlo en la mayoría de los casos ya es muy tarde. Es por ello, que al tener una herramienta al alcance de todos es más fácil para el usuario realizarle la prueba y para los

especialistas poder llevar un control. Por último, en el Anexo 11 se confirma que el despliegue de la solución que se ha desarrollado en el presente trabajo de investigación está validado por una especialista en neurología, quien trabaja con pacientes que presentan la EP. Además, se siguen recolectando muestras de imágenes del examen de escritura, ya que estos datos se guardan en una base de datos en tiempo real.

Es importante recalcar, que la base de datos creada en esta tesis, resultados más importantes y códigos relevantes para el presente trabajo de investigación se encuentran en el siguiente repositorio: <https://github.com/virginiamonroy/Tesis-VMM-2020>. Además, en este repositorio se encuentran fotos que validan como fue la recolección de los datos en la ciudad de Arequipa, dos videos (en el primero se muestra cómo es que tiene que ser llenado el examen de escritura y en el segundo, se muestra cómo funciona la interacción con el asistente virtual).

Con todos los resultados y despliegues mostrados anteriormente, se puede concluir que los mejores resultados se obtienen de las combinaciones de CNN + ML tanto para meandros como para espirales (*Accuracy* = 99.9%). Sin embargo, se puede rescatar que obteniendo puntos característicos del descriptor SIFT con $k = 50$, posteriormente clasificado con SVM, también se obtienen resultados aceptables (*Accuracy* = 95% para meandros y *Accuracy* = 90% para espirales). Por lo tanto, todos los experimentos realizados en esta investigación obtienen resultados comparables al estado del arte e inclusive bajo ciertos escenarios mejores, como es el caso del estudio de Clayton et al., (2016), dado que, en ese trabajo de investigación, utilizando técnicas de Deep Learning como LeNet obtuvieron (*Accuracy*=90.585). Sin embargo, en estos casos para detectar una enfermedad se necesita la mayor precisión posible, por lo que CNN + ML sería el modelo para elegir y se concluye que mediante la implementación de un modelo de visión computacional se logra determinar el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona, asimismo gracias al despliegue que se realizó, este puede ser aplicado en tiempo real.

Asimismo, respecto a las hipótesis específicas planteadas se puede afirmar lo siguiente: Se logra confirmar la primera hipótesis, la cual planteaba que la identificación y creación de bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson permite el desarrollo del modelo. Además, la creación de una base de datos que fue resultado de un examen de escritura aplicado a los miembros de la Asociación Activate Liga contra el Parkinson Arequipa y a personas sanas. Seguidamente, para la construcción de la base de datos se tomaron fotos al mencionado examen con una cámara de 64 px a una distancia de 13 cm,

con un proceso de tomado de fotos estandarizado para todas ellas. Además, la base de datos, en este caso balanceada entre personas que presentan la EP y personas que no, permitió la construcción y desarrollo de un nuevo modelo, con lo cual confirmamos nuestra primera hipótesis. (Ver detalle en el apartado 4.1 Adquisición del *Dataset*)

Respecto a la segunda hipótesis, la cual planteaba que la implementación de técnicas de procesamiento de imágenes, como filtros de mediana, aumento de brillo, contraste, entre otros, sobre imágenes de manuscritos influye positivamente para poder aumentar la calidad de la imagen, se puede afirmar, dado que en el preprocesamiento de la imagen uno de los pasos claves fue poder realizar técnicas de mejora de imagen con filtros como poder darle una mejor calidad a la imagen y con todo ello se obtuvieron mejores resultados incrementando la precisión del modelo de 50% a más del 90%. (Ver detalle en el apartado 4.2 Fase de Preprocesamiento)

Del mismo modo, la tercera hipótesis, la cual planteaba que la implementación de técnicas de visión computacional favorece para poder extraer vectores característicos descriptores como color, forma o textura de las imágenes, se puede afirmar, dado que en el proceso de extracción de características se obtuvieron vectores característicos, los cuales fueron entradas para los modelos de clasificación. Es importante mencionar que con el descriptor SIFT y SURF se obtuvieron los mejores resultados a diferencia de HOG, ya que este último no encontró muchos puntos característicos en las imágenes de la base de datos. (Ver detalle en el apartado 4.3 Extracción de características)

Además, la cuarta hipótesis, la cual planteaba que la utilización de técnicas de machine learning influye positivamente en la clasificación de la imagen de la escritura de una persona respecto a si pertenece a una persona con diagnóstico de la EP, se puede afirmar, dado que modelos de machine learning como SVM y CNN fueron necesarios e influyeron positivamente en la clasificación de si una persona presenta o no la EP con un 99% de precisión. (Ver detalle en el apartado 4.4 Modelo y Clasificación)

Finalmente, la quinta hipótesis, la cual planteaba que la utilización de métricas influye positivamente en la evaluación de la imagen de la escritura de una persona respecto a si pertenece a una persona con diagnóstico de la EP, se puede afirmar. Dado que, modelos de machine learning como SVM, RF, KNN y CNN fueron necesarios e influyeron positivamente en la evaluación de si una persona presenta o no la Enfermedad de Parkinson con resultados exitoso con más de 90% de precisión en su mayoría. (Ver detalle en el apartado CAPÍTULO 5: ANÁLISIS Y DISCUSIÓN DE RESULTADOS)

Capítulo 6: Conclusiones y Recomendaciones

En el presente capítulo se va a dar las conclusiones y recomendaciones generales, las cuales han sido identificadas después de haber culminado el trabajo de investigación.

6.1 Conclusiones

Después de haber identificado el problema al inicio de la presente tesis se pudo identificar que existe una gran problemática que no solo afecta de manera local sino lo hace de manera mundial. Además, se mencionó en los antecedentes que existen algunas investigaciones que proponen lo siguiente: la EP puede ser predicha a través de la escritura utilizando técnicas de Inteligencia Artificial, a diferencia de las mencionadas investigaciones en este trabajo se prueban distintos frameworks para demostrar este postulado y así mismo se plantea un despliegue para poner en producción el modelo de clasificación creado.

Existen varios factores que hacen que muchas personas traten de buscar una solución o cura para combatir la Enfermedad de Parkinson. Primero, la EP es la segunda enfermedad neurológica que más afecta a la población. Segundo, actualmente no existe un método certero con el cual se pueda detectar si una persona presenta o no la EP. Tercero, actualmente no existe una cura para la EP por lo cual si una persona padece esta enfermedad está condenada a vivir con esta. Cuarto, las personas que presentan la EP llevan una baja calidad de vida y no solo le afecta a dicha persona sino también a todo su entorno social.

Bajo estos factores presentados de manera resumida, surgió la necesidad de encontrar alguna forma para poder ayudar en la detección previa de esta enfermedad, ya que, si bien no tiene cura, si es detectada a tiempo se puede iniciar un tratamiento con el cual se puede postergar la aparición de dicha enfermedad y de ese modo tener una buena calidad de vida.

Como primer paso, se obtuvo una base de datos que contenía imágenes de manuscritos de personas que presentaban la EP o no, de un repositorio que fue recolectado por la Facultad de Medicina Botucatu, Universidad Estatal de São Paulo, Brasil, el cual contiene una muestra de 632 imágenes etiquetadas respectivamente para meandros y espirales. Asimismo, se construyó una nueva base de datos con 1080 imágenes de manuscritos de personas que presentaban la EP o no, a través de exámenes escritos que se le aplicó a una muestra de 27 personas de la Asociación Actívate Liga contra el Parkinson Arequipa para la recolección de personas que presentan la EP y se aplicó el mismo examen a 27 personas de diferentes edades que no presentan la EP. Por lo tanto, se cumple el primer objetivo que es identificar y crear bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson.

El objetivo de implementar técnicas de procesamiento de imágenes a imágenes de manuscritos para poder aumentar la calidad de la imagen se cumplió, dado que inicialmente se pretendió construir los modelos predictivos sin aplicar preprocesamiento a las imágenes y se obtuvieron modelos que presentaban *overfitting* y sus métricas en test no pasaban del 50%. Por lo cual, antes de la construcción del modelo se realizó un tratamiento a las imágenes, el cual consistió en escala de grises, filtro de mediana, aumento de contraste y de brillo. Con este preprocesamiento aplicado a las imágenes se obtuvieron resultados de más del 90% de precisión.

Después del preprocesamiento, se utilizaron técnicas de descriptores para poder obtener los vectores característicos representativos de una imagen. Se utilizaron las técnicas SIFT, SURF, ORB y HOG. Es así que se cumplió el tercer objetivo, implementar técnicas de visión computacional para obtener vectores característicos que describan el color, forma o textura de las imágenes.

Seguidamente, se hizo uso de técnicas de *Machine Learning* como SVM, RF y KNN para la creación del modelo clasificador y *Deep Learning* con *frameworks* como VGG16, VGG19, Inception, ResNet y LeNet para el entrenamiento del modelo y para la creación de este. Finalmente, se predijo con éxito, si bajo ese síntoma de la escritura anómala de un paciente con EP, una persona tiende a presentar o no la EP. Con lo cual, se cumplió el cuarto objetivo, utilizar técnicas de *machine learning* para poder clasificar si la imagen del manuscrito pertenece a una persona con diagnóstico positivo de la enfermedad de Parkinson. Cabe mencionar que en las redes utilizadas no se han regularizado los pesos, ya que se utilizaron los pesos preentrenados de la competencia ImageNet.

Asimismo, se cumplió el quinto objetivo, utilizar métricas que permitan evaluar la eficiencia de la clasificación si la imagen del manuscrito pertenece a una persona con diagnóstico de la enfermedad de Parkinson, dado que se utilizaron las métricas *Accuracy*, *Precision*, *Recall* y F1 para evaluar que tan bien estaba clasificando el modelo de predicción. La precisión del modelo de clasificación tanto para meandros como para espirales es 99.9%.

En este trabajo de tesis se ha desarrollado de manera teórica y práctica la implementación de un modelo de visión computacional para el prediagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona, es un prediagnóstico, ya que la única forma de que esta enfermedad sea diagnosticada es a través de una tomografía. A través de técnicas de visión computacional se procesaron imágenes de escrituras extraídas de pacientes sanos y

de pacientes que presentan EP, para extraer vectores característicos a partir de dichas imágenes, posteriormente crear los algoritmos de predicción y a través de las métricas se evaluó cual era el mejor, lo cual valida el cumplimiento del objetivo general.

Es importante mencionar que, con los resultados obtenidos se espera poder contribuir con este problema de talla mundial, ya que podríamos cambiar la vida de muchas personas, no solo a aquellos que podrían padecer la enfermedad sino también a su entorno social. Respecto a los falsos positivos, cuando estos eran detectados, primero se procedió a eliminar imágenes que ocasionaban mucho ruido al experimento, segundo estas pasan por un filtro para la clasificación final, se toma un examen completo con más imágenes para los trazos y de ese modo evitar los falsos positivos.

Finalmente, como aporte adicional al presente trabajo de investigación, se ha creado un despliegue en Facebook, para que más personas puedan llegar a este, que sirve para poder aplicar el modelo predictivo a través de un examen en línea en tiempo real, recolectar información de las personas para poder llegar a ellas a través de una acción proactiva, y también se ha contactado con la presidente de la Asociación en Arequipa y dos neurólogas especialistas en el tema para poder dejar sus datos y así los pacientes se puedan contactar con ellas.

Además, con los dos despliegues que se realizaron en el presente trabajo de investigación, se tiene por objetivo llegar de una manera más rápida a personas de todo el país, para poder tomar acción preventivamente, ya que las estadísticas indican que esta enfermedad está en aumento y como se explicó previamente si es detectada a tiempo, los efectos de la misma se pueden postergar. Por último, dejo abierta la posibilidad de que este trabajo pueda ser mejorado conforme pasan el tiempo, ya que las técnicas de visión computacional están en aumento.

6.2 Recomendaciones

A partir de lo concluido, se podría determinar con certeza que la implementación de un sistema de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a través de la escritura de una persona funciona de manera efectiva incluso más que otros métodos los cuales ya están implementados. Además, como recomendaciones técnicas adicionales se propone aumentar la cantidad de imágenes obtenidas en esta tesis para poder entrenar con más muestras al modelo, probar nuevas técnicas de *Deep Learning* como el uso de GANs o probar con otras arquitecturas como GoogLeNet o AlexNet. Por último, se podrían aplicar otras

técnicas de visión computacional, ya que como estamos en un constante avance en la tecnología, cada vez se crean nuevas técnicas o se optimizan las utilizadas en este trabajo de tesis.

En el caso de las CNNs se usó una función sigmoideal para la clasificación, además, se utilizó una regla de 0.7, es decir, si la probabilidad es mayor a esta se etiqueta como clase positiva y en su defecto clase negativa. Esta regla definida puede ser modificada a fin de obtener mejor rendimiento en la clasificación. Asimismo, los resultados de la clasificación pueden ser mejorados si se gradúa el valor del bias en una red neuronal.

Adicionalmente, se recomienda el desarrollo de un modelo multiclase que permita prediagnosticar el nivel de la enfermedad en el que se encuentra una persona que ha sido prediagnosticada con EP.

Asimismo, una recomendación para aquellas personas que quiera dedicarse a la investigación en este tema, sería primero conocer de manera específica todo lo relacionado a la Enfermedad de Parkinson, ya que de ese modo se logra tener un interés mayor para realizar el proyecto, puesto que con los resultados se pueden beneficiar a muchas personas. Finalmente, la aplicación de este tipo de trabajos en el Perú podría cambiar el modo de pensar de todos nosotros y generar que se implementen nuevas soluciones a diferentes problemáticas no solo en el rubro de salud sino en educación, cultura, gobierno, entre otros que vayan de la mano con la tecnología, ya que está es el futuro.

Referencias

- Alpaydin, E. (2016). *Machine Learning*. Massachusetts: The MIT Press
- Jakhar, P. (4 de marzo de 2020). Coronavirus: las innovadoras tecnologías que está utilizando China para combatir el covid-19 (y las preocupaciones que plantean). *BBC NEWS*. Recuperado de: <https://www.bbc.com/mundo/noticias-51736635>
- Bay, H., Tuytelaars, T & Van Gool, L. (mayo de 2006). *SURF: Speeded Up Robust Features*. Conferencia europea sobre visión artificial (págs. 404-417). Springer, Berlín, Heidelberg.
- Chang, D. Alban Hidalgo, M & Hsu, K. (2014). *Diagnosing Parkinson's Disease From Gait*. Stanford.
- Chollet, F. (2018). *Deep Learning with Python*. New York: Manning.
- ¿Cómo superar el Parkinson? Consejos de una paciente que venció las dificultades. (23 de diciembre de 2019). CORREO. Recuperado de <https://diariocorreop.e/edicion/arequipa/como-superar-el-parkinson-consejos-de-una-paciente-que-vencio-las-dificultades-928439/>
- Cowley, J. (2018). Redes neuronales convolucionales. USA: IBM. Recuperado de: <https://developer.ibm.com/es/technologies/artificial-intelligence/articles/cc-convolutional-neural-network-vision-recognition/>
- Dalal, N. & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. Conferencia de la Sociedad de Computadoras sobre Visión por Computador y Reconocimiento de Patrones (CVPR'05), San Diego, CA, EE. UU., EE. UU.
- Davida, B. (2018). Bag of Visual Words in a Nutshell. USA: *Towards Data Science*. Recuperado de: <https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb>
- De la Fuente, S. (2011). Regresión logística. Madrid: Universidad Autónoma de Madrid, 1-29.
- Deloitte. (2015). Encuesta Global de Consumidores Móviles 2015 – Edición. Colombia: Deloitte. Recuperado de: <https://www2.deloitte.com/co/es/pages/technology-media-and-telecommunications/articles/encuesta-global-de-consumidores-moviles---2015---deloitte-colomb.html>

- De Stefano, C., Fontanellal, F., Impedovo, D., Pirlo, G. & Scotto, A. (2017). A Brief Overview on Handwriting Analysis for Neurodegenerative Disease Diagnosys. Dipartimento di Ingegneria Elettrica e dell'Informazione (*DIEI*) (pp. 9-16).
- Deng, H. (2018). An introduction to Random Forest. USA: *Towards Data Science*. Recuperado de: <https://towardsdatascience.com/random-forest-3a55c3aca46d>
- EcuRed. (2020). Agente transmisor [Figura]. Recuperado de: https://www.ecured.cu/S%C3%ADndrome_parkinsoniano
- Efisioterapia. (2012). Exploración motora. [Figura]. Recuperado de: <https://www.efisioterapia.net/articulos/estudio-fisioterapia-enfermedad-parkinson>
- EnciclopediaSalud. (2020). Definiciones. Recuperado de <https://www.encyclopediasalud.com/>
- García, S., López, B. & Meza, E., Villagómez, A. & Vázquez, R. (2010). Breve reseña histórica de la enfermedad de Parkinson. De la descripción precipitada de la enfermedad en el siglo XIX, a los avances en Biología Molecular del padecimiento. *Medigraphic*, 357 - 358 - 359.
- Gil Martín, M., Montero, J. & San Segundo, R. (2019). Parkinson's Disease Detection from Drawing Movements Using Convolutional Neural Networks. *Electronics*, 8(8), 907.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. Massachusetts :MIT Press. Recuperado de <http://www.deeplearningbook.org>
- González L. (2018) Aprendizaje Supervisado: Decision Tree Clasification. *Aprende IA*. Recuperado de <http://ligdigonzalez.com/aprendizajesupervisado-decision-tree-classification/>
- Harrington, P. (2012). *Machine Learning in Action*, New York, EE.UU: Manning Publications.
- Harrison, O. (2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Recuperado de: [https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761#:~:text=KNN%20works%20by%20finding%20the,in%20the%20case%20of%20regression\)](https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761#:~:text=KNN%20works%20by%20finding%20the,in%20the%20case%20of%20regression)).
- Haykin, S. (2009). *Machine Learning in Action*. (3ra. Ed.). Hamilton, Canada: Pearson Education.

- He, K., Zhang, X., Ren, S. & Sun, J. (2016). Aprendizaje residual profundo para el reconocimiento de imágenes. En Actas de la conferencia IEEE sobre visión por computadora y reconocimiento de patrones (págs. 770-778).
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la investigación* (5ta. Ed.). Santa Fe: Mc GRAW HI. Recuperado de <https://upvv.clavijero.edu.mx/cursos/LEB0742/documentos/Metodologiadelainvestigacion.pdf>
- Instituto Nacional de Ciencias Neurológicas. (2018). Crece número de casos de parkinson en personas jóvenes. Perú: *Ministerio de Salud*. Recuperado de: <http://www.incn.gob.pe/index.php/40noticias-actuales/549-crece-numero-de-casos-de-parkinson-en-persona-jovenes>
- Joshi, R. (2016). Exactitud, precisión, recuperación y puntuación F1: interpretación de las medidas de rendimiento. India: *Exsilio Solutions*. Recuperado de: <https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#:~:text=Precision%20%2D%20Precision%20is%20the%20ratio,the%20total%20predicted%20positive%20observations.&text=F1%20score%20%2D%20F1%20Score%20is,and%20false%20negatives%20into%20account>
- Li, J., Qiu, T., Wen, C., Xie, K., & Wen, F. Q. (2018). Robust face recognition using the deep C2D-CNN model based on decision-level fusion. *Sensors*, 18(7), 2080.
- Marti, A. (2020). Motorola Moto G9 Plus: la "superbatería" llega al Moto G9 más potente hasta la fecha. [Figura]. Recuperado de <https://www.xataka.com/moviles/motorola-moto-g9-plus-caracteristicas-precio-ficha-tecnica>
- Narkhede, S. (2018). Entendiendo AUC-Curva ROC. USA: *Towards Data Science*. Recuperado de: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>
- Rodríguez Riaño, J., & Baquero Catellanos, S. (2011). Escritura en enfermedad de Parkinson (EP): Revisión sistemática de literatura. *Acta Colombiana de Psicología*, Vol. 14, no. 2 (jul.-dic. 2011); p. 17-26.
- Kalana, R. & Sunny, L. (2018). Detection of Parkinson's Disease Through Static Analysis of Handwriting and Character Recognition. *IRJET*, 05(05), 3676-3680.

- Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
- Nacional Institute of Neurological Disorders and Stroke. (2016). Enfermedad de Parkinson: Esperanza en la investigación. USA: *National Institute of Neurological Disorders and Stroke*. Recuperado de https://espanol.ninds.nih.gov/trastornos/parkinson_disease_spanish.htm
- Organización Mundial de la Salud. (2016). Parkinson y yo. Recuperado de: <http://terapiaparkinson.com/testimonialview/organizacion-mundial-de-la-salud-oms/>
- OMS. (2020). Preguntas y respuestas sobre la enfermedad por coronavirus (COVID-19). Recuperado de: <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses>
- Parkinson's News Today. (2018). Parkinson's Disease Statistics. Pensacola: *Parkinson's News Today*. Recuperado de: <https://parkinsonsnewstoday.com/parkinsons-disease-statistics/>
- Pereira, C., Pereira D., Da Silva, F., Slike T., Hook C. & Weber, C., Pereira, L., Papa, J. (2015). A Step Towards the Automated Diagnosis of Parkinson's Disease: Analyzing Handwriting Movements. Simposio internacional de IEEE sobre sistemas médicos basados en computadoras (págs. 171-176). IEEE, Sao Carlos, Brasil.
- Pereira, C., Weber, S., Hook, C., Rosa, F. & Papa, J. (2016). Deep Learning-aided Parkinson's Disease Diagnosis from Handwritten Dynamic. Simposio internacional de IEEE SIBGRAPI sobre Gráficos, Patrones e Imágenes (SIBGRAPI), Sao Paulo, Brasil.
- Plataforma digital única del Estado Peruano. (2020). Gobierno lanza nueva versión de app "Perú en tus manos" para advertir a los ciudadanos sobre las zonas con mayor probabilidad de contagio. Perú: *Plataforma digital única del Estado Peruano*. Recuperado de: <https://www.gob.pe/institucion/pem/noticias/150943-gobierno-lanza-nueva-version-de-app-peru-en-tus-manos-para-advertir-a-los-ciudadanos-sobre-las-zonas-con-mayor-probabilidad-de-contagio>

- Plataforma digital única del Estado Peruano. (2020). Encuesta Nacional COVID-19. Perú: *Plataforma digital única del Estado Peruano*. Recuperado de: <https://www.gob.pe/8979-encuesta-nacional-covid-19>
- Ponce, P. (2010). *Inteligencia Artificial con aplicaciones a la Ingeniería*. (1ra. Ed.). México D.F.: Alfaomega.
- Real Academia Nacional de Medicina de España (2012). Signo y Síntoma en Medicina. España: *Traducción médica, técnica y literaria*. Recuperado de: <https://traduccionmedicatecnicaliteraria.wordpress.com/2016/11/05/signo-y-sintoma-en-medicina/>
- Regueiro, S. & Leurona E. (2014). Ilustración de sustancia negra en la Enfermedad de Parkinson. [Figura]. Recuperado de: <https://www.elsevier.es/es-revista-farmaciaprofesional-3-articulo-curso-basico-formacion-terapeutica-pacientesX021393241431692X>
- Rich, E. & Knight, K. (1991). *An Introduction to Artificial Intelligence*. (2da. Ed.). McGraw Hill Higher Education.
- Ruble, E., Rabaud, V., Konolige, K. y Bradski, G. (2011). ORB: Una alternativa eficaz a SIFT o SURF. Conferencia internacional sobre visión artificial IEEE (págs. 2564-2571).
- Russel, S. & Norvig, P. (2010). *Artificial Intelligence a modern Approach*. (3ra. Ed.). New Jersey: Alan Apt. Recuperado de <https://www.cin.ufpe.br/~tfl2/artificial-intelligence-modern-approach.9780131038059.25368.pdf>
- Saxena, P. (2020). Redes residuales (ResNet): aprendizaje profundo. India
- Scikit-image. (s.f.). Histogram of Oriented Gradients [Figura]. Recuperado de: https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. Conferencia arXiv preprint arXiv:1409.1556.
- Sucar, L. & Gómez, G. (1991). Visión Computacional. Instituto Nacional de Astrofísica, Óptica y Electrónica. México. Recuperado de: <https://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2014). Going deeper with convolutions. Actas de la Conferencia IEEE sobre Visión por Computador y Reconocimiento de Patrones (CVPR), 2015, págs. 1-9
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. (2da. Ed.). Washington: Springer. Recuperado de https://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf
- Swamynathan, M. (2017). *Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python*. Karnataka: Todd Green. Recuperado de <https://tanthiamhuat.files.wordpress.com/2018/04/mastering-machine-learning-with-python-in-six-steps.pdf>
- Tagle, P. (2005). Historia de la enfermedad de Parkinson. Publicaciones Medicina, 1-3. Recuperado de <http://publicacionesmedicina.uc.cl/cuadernos/2005/HistoriaParkinson.pdf>
- Zornoza, J. (2020). *Random Forest Explained*. Recuperado de: <https://towardsdatascience.com/random-forest-explained-7eae084f3ebe>

Anexos

Anexo 1: Matriz de Contingencia

TÍTULO DE LA TESIS	Diseño de un sistema de visión computacional para el pre-diagnóstico de la Enfermedad de Parkinson a partir de la escritura de una persona			
PROBLEMA PRINCIPAL	OBJETIVO PRINCIPAL	HIPÓTESIS PRINCIPAL	VARIABLE	MÉTODO
¿Es posible utilizar técnicas de visión computacional para el prediagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona?	Implementar un modelo de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona	Mediante la implementación de un modelo de visión computacional se logra determinar el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona	Dependiente: Pre-diagnóstico de la Enfermedad de Parkinson Independiente: Imagen de escritura de una persona	Tipo de Investigación: De carácter experimental Alcance de la Investigación: De carácter explicativo, orientado al descubrimiento de factores origen del fenómeno
PROBLEMAS ESPECÍFICOS	OBJETIVOS ESPECÍFICOS	HIPÓTESIS ESPECÍFICOS	VARIABLE	MÉTODO
¿Existen bases de datos disponibles de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson?	Identificar y crear bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson	La identificación y creación de bases de datos de imágenes de escritura de pacientes que presenten la enfermedad de Parkinson permite el desarrollo del modelo		
¿Qué técnicas de procesamiento de imágenes se pueden aplicar a imágenes de manuscritos para poder	Implementar técnicas de procesamiento de imágenes a imágenes de manuscritos para poder	La implementación de técnicas de procesamiento de imágenes a imágenes de manuscritos influye positivamente para poder	Dependiente: Calidad de imagen Independiente: Técnicas de preprocesamiento de imágenes	Tipo de Investigación: De carácter experimental Alcance de la Investigación: De carácter explicativo, orientado al

aumentar la calidad de la imagen?	aumentar la calidad de la imagen	aumentar la calidad de la imagen		descubrimiento de factores origen del fenómeno
¿Qué técnicas de visión computacional existen para volver vectores característicos características como color, forma o textura de las imágenes?	Implementar técnicas de visión computacional para obtener vectores característicos que describan el color, forma o textura de las imágenes	La implementación de técnicas de visión computacional favorece para poder extraer vectores característicos características como color, forma o textura de las imágenes		
¿Qué tipos de técnicas de machine learning son posibles de aplicar para poder clasificar si la imagen del manuscrito pertenece a una persona con diagnóstico de la enfermedad de Parkinson?	Utilizar técnicas de machine learning para poder clasificar si la imagen del manuscrito pertenece a una persona con diagnóstico positivo de la enfermedad de Parkinson	La utilización de técnicas de machine learning influye positivamente en la clasificación de la imagen de la escritura de una persona respecto a si pertenece a una persona con diagnóstico de la enfermedad de Parkinson	Dependiente: Persona con diagnóstico de la Enfermedad de Parkinson Independiente: Técnicas de Machine Learning	Tipo de Investigación: De carácter experimental Alcance de la Investigación: De carácter explicativo, orientado al descubrimiento de factores origen del fenómeno
¿Qué métricas permiten evaluar la eficiencia de la clasificación si la imagen del manuscrito pertenece a una persona con diagnóstico de la enfermedad de Parkinson?	Utilizar métricas que permitan evaluar la eficiencia de la clasificación si la imagen del manuscrito pertenece a una persona con diagnóstico de la enfermedad de Parkinson	La utilización de métricas influye positivamente en la evaluación de la imagen de la escritura de una persona respecto a si pertenece a una persona con diagnóstico de la enfermedad de Parkinson		

Anexo 2: Validez Instrumento de medida

FICHA DE VALIDACIÓN DE INSTRUMENTO

Título del Proyecto

Diseño de un sistema de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona

Objetivo General: Implementar un modelo de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona.

Nombre del investigador: Virginia del Pilar Monroy Malca

Nombre del experto: Koni Katerin Mejía Rojas

Especialidad: Neurología

Instrucciones: Determinar si el instrumento de Examen de escritura reúne los indicadores señalados y evaluar en la escala de Excelente, muy bueno, bueno, regular y malo.

Nº	Indicadores	Definición	Excelente	Muy Buena	Buena	Regular	Mala
1	Claridad y precisión	Las figuras son claras	X				
2	Validez	Las imágenes ya han sido validadas en un artículo científico publicado en (ScienceDirect)	X				
3	Organización	La estructura es adecuada. Comprende la presentación, agradecimiento, datos e instrucciones	X				
4	Confiabilidad	El instrumento es confiable porque es una adaptación de un examen de escritura ya validado	X				
5	Marco de referencia	Las imágenes han sido colocadas de acuerdo al marco de referencia del usuario: trazos	X				
6	Extensión	El número de imágenes no es excesivo y está en relación a las variables, dimensiones e indicadores del problema.	X				
7	Inocuidad	Las imágenes no constituyen riesgo para el usuario.	X				

En consecuencia, el instrumento puede ser aplicado


Firma del experto
DNI: 43280792

Anexo 3: Documento emitido por la Universidad ESAN – Carta de Presentación



Santiago de Surco, 08 de setiembre de 2020

Señorita

Christine Allison Jeyachandran

Presidenta

ASOCIACION ACTIVATE AREQUIPA LIGA CONTRA EL PARKINSON

Presente. -

De mi consideración,

Tengo a bien dirigirme a usted para saludarla cordialmente y a la vez, presentarle a la señorita Virginia del Pilar Monroy Malca, bachiller de la carrera *de Ingeniería de Tecnologías de Información y Sistemas* de nuestra casa de estudios, quien está desarrollando la tesis de titulación denominada '*Diseño de un sistema de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona*'.

Motivo por el cual, solicito su autorización para que la Srta. Monroy pueda efectuar la recolección de datos de su representada. Cabe mencionar que la bachiller tiene el compromiso de mantener la confidencialidad de la información obtenida.

Agradeciendo anticipadamente la atención a la presente, quedo de usted.

Atentamente,

JAVIER DEL CARPIO GALLEGOS

Decano

Facultad de Ingeniería

Anexo 4: Documento de Aceptación de Recolección de Datos



Arequipa, 08 de setiembre de 2020

Autorización:

Yo Christine Allison Jeyachandran, presidenta de la ASOCIACION ACTIVATE AREQUIPA LIGA CONTRA EL PARKINSON identificada con Carnet de Extranjería número: 000756817 autorizo a Virginia del Pilar Monroy Malca con documento de identidad número: 74043491 a que tenga acceso a los datos personales nombre y apellidos de los miembros de la ASOCIACION ACTIVATE AREQUIPA LIGA CONTRA EL PARKINSON. Asimismo, doy autorización para que pueda efectuar la recolección de datos a través de un EXAMEN DE ESCRITURA que se le aplicará a los miembros de la Asociación personalmente.

Atentamente.

Christine Allison Jeyachandran

CE: 000756817

Anexo 5: Carta de presentación para los miembros de la Asociación

Arequipa, 27-08-2020

ASUNTO: Ayuda para pre-diagnosticar la enfermedad del Parkison

Con mucho agrado, a través de **Christine Jeyachandran**, tengo el gusto de conocerlos.

Mi nombre es **Virginia del Pilar Monroy Malca**, nací en Arequipa y viví en esta hermosa ciudad hasta los 17 años. A esta edad me fui a estudiar a Lima en la Universidad ESAN en la cual obtuve el grado de **bachiller en Ingeniería de Tecnologías de Información y Sistemas**.

A fin de optar mi grado de bachiller realicé una tesis denominada **"Diseño de sistema de visión computacional para el pre-diagnóstico de la Enfermedad de Parkinson a partir de la escritura"**, la cual fue **aprobada** por la universidad. Esta misma tesis la estoy perfeccionando y la voy a presentar para mi grado profesional.

Por lo cual, **necesito el apoyo de todos ustedes** integrantes de la Asociación Actívate - Liga de Lucha contra el Parkinson de Arequipa. Entre los objetivos, que tiene su institución está promover la educación y lucha contra el PARKINSON; el cual es un fin muy loable. Sin embargo, mi trabajo plantea ir un poco más allá para luchar y **hacer más sobrellevadera esta enfermedad, previniendo y/o detectando de manera anticipada a posibles personas que van a desarrollar esta enfermedad a través del tiempo**, realizando un prediagnostico a través de la escritura.

Tal vez, no me toque en esta oportunidad apoyarlos de manera directa a ustedes, pero nosotros **juntos sí se puede ayudar a personas que van a pasar los síntomas y secuelas de la misma**. ¿Qué bonito fuere que con este pequeño granito de arena y con los aportes que hace su institución y otros, los **nuevos pacientes puedan anticiparse y hacer terapias al comienzo de la enfermedad?** lo cual conllevaría a que su ritmo de vida, vida familiar, trabajo, y demás sea más **llevadera**.

Para esto necesito que como ustedes ya son personas diagnosticadas con esta enfermedad me **apoyen en el llenado de una ficha anónima que consta de realizar trazos manualmente con un lapicero de color azul**; estos serán analizados por una computadora para encontrar patrones en común que diagnostiquen la Enfermedad de Parkinson, dado **que la computadora ve lo que el ojo humano no distingue**. De ser validados estos datos de manera correcta, lograremos que aplicado este mismo tipo de ficha a personas en edad temprana podamos ayudar a ellas a saber si son propensos o no a tener esta enfermedad para que puedan afrontarla de la manera correcta y retardar los efectos de la misma.

Estoy segura desde ya que es deber de las personas trascender en la vida y esta sería una forma de hacerlo. Cuento con su apoyo gracias.

Virginia del Pilar Monroy Malca

Bachiller en Ingeniería de Tecnologías de Información y Sistemas

DNI: 74043491

Celular: 953548875

Correo: virginiamonroymalca@gmail.com

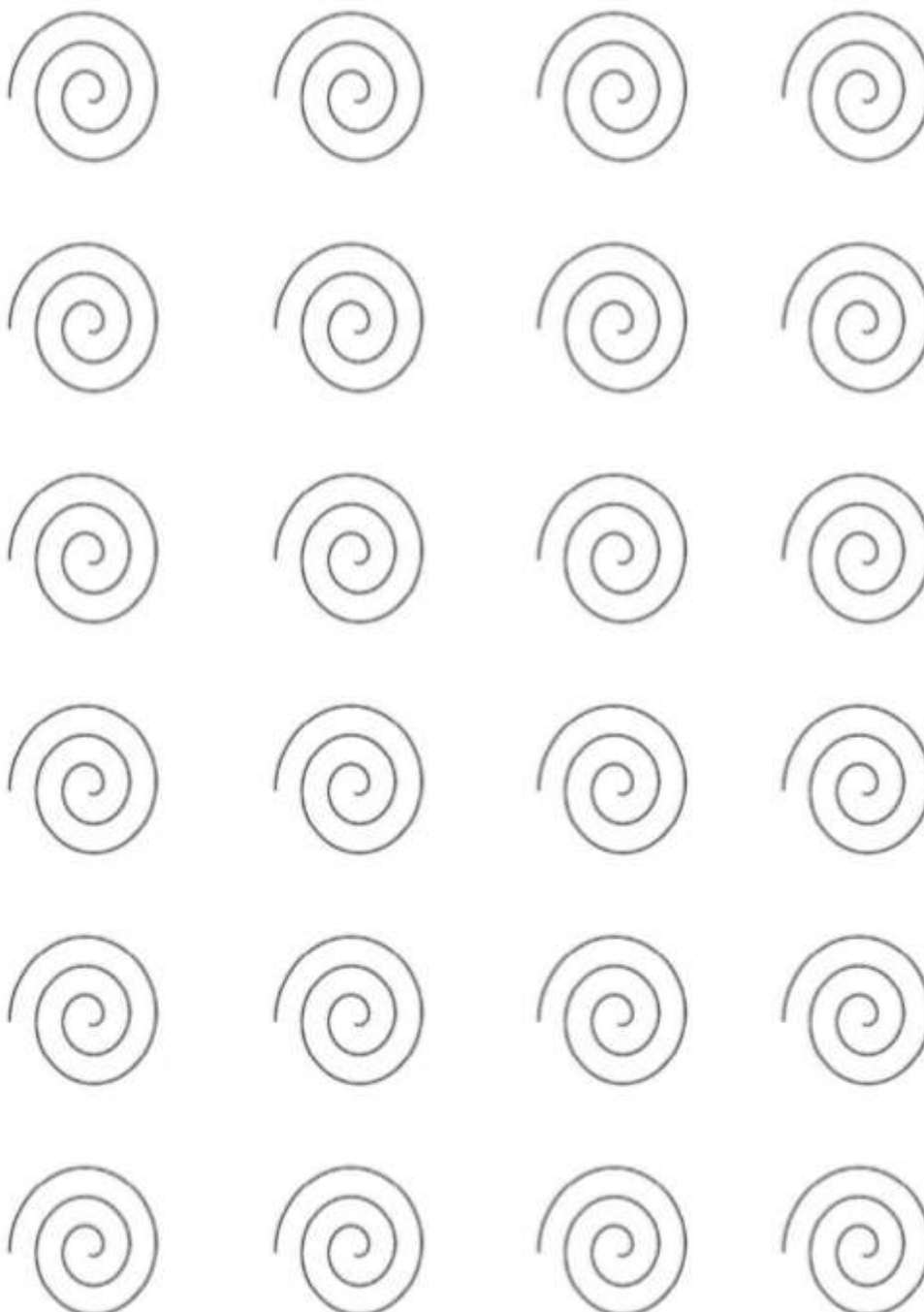
Anexo 6: Examen de Escritura

EXAMEN DE ESCRITURA

Edad: ____

Mano Dominante: () Derecha () Izquierda

Espirales:

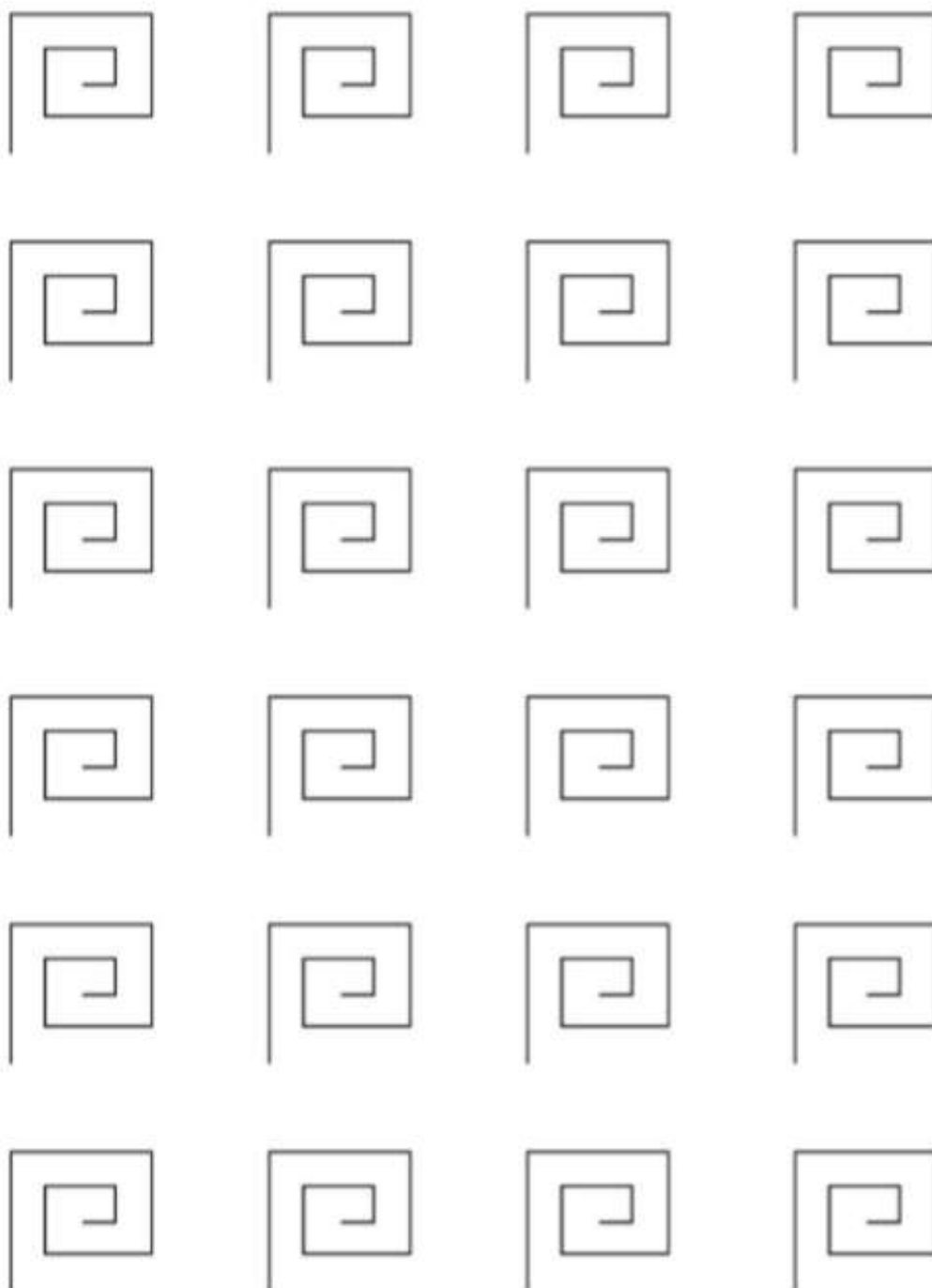


EXAMEN DE ESCRITURA

Edad: ____

Mano Dominante: () Derecha () Izquierda

Meandros:



Anexo 7: Instrucciones de llenado del Examen de Escritura

INSTRUCCIONES PARA EL LLENADO DEL EXAMEN DE ESCRITURA

Previo al llenado del examen tiene usted que tener en cuenta lo siguiente:

1. Abra usted el fólder, en el cual encontrará las fichas del examen de escritura.
2. De preferencia realice el llenado del examen de escritura en un lugar donde tenga buena iluminación.
3. El examen de escritura consta de 2 hojas, una con figuras de eespirales y la otra con figuras de meandros **(ambas deben ser llenadas)**.
4. En caso se malogre una de las primeras hojas, en el sobre encontrará una ficha adicional **(de otras 2 hojas)**, las cuales deben ser llenadas solo en caso de que se malogren las dos primeras.
5. Realizar los trazos con un lapicero azul **(obligatorio)**.
6. Cada dibujo tiene que ser en un solo trazo sin soltar el lapicero y así sucesivamente.
7. El trazo tiene que ser natural sin importa que no sea perfecto
8. No importa el tiempo que tarde en terminar el examen de escritura **(tómese su tiempo)**.
9. No usar otro tipo de instrumento **(ejemplo regla)**.
10. Debe realizar los trazos sin ayuda de otra persona.
11. Solo puede ser llenado por la persona que presente la Enfermedad de Parkinson.
12. El sábado 12 del presente mes, procederé a recoger las fichas debidamente llenadas.

Nota: En caso tenga usted alguna duda para el llenado por favor comunicarse con mi persona a través del siguiente número: **953548875**

Anexo 8: Noticias Actívate – pag. 7



Virginia, quien es bachiller en Ingeniería de Tecnologías de Información y Sistemas de la Universidad Esan se encuentra desarrollando un proyecto de investigación con el que pretende detectar de manera temprana la Enfermedad de Parkinson a través de la escritura aplicando la Inteligencia Artificial.

Para lo cual, como parte de su investigación viene aplicando exámenes de escritura de manera anónima con algunos integrantes de nuestra asociación habiéndose contactado con los mismos respetando los protocolos de distanciamiento social dentro del marco de la emergencia sanitaria. Este proyecto, que tiene mucho potencial ayudará a detectar de manera temprana, en personas jóvenes y adultas, esta enfermedad para que así estas puedan iniciar las terapias y/o tratamiento médico de manera oportuna.

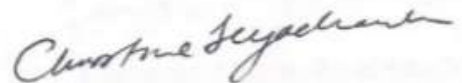
Anexo 9: Autorización para proporcionar nombre y teléfono – Christine Jeyachandran

Arequipa, 02 de diciembre de 2020

Autorización:

Yo Christine Allison Jeyachandran, presidenta de la ASOCIACION ACTIVATE AREQUIPA LIGA CONTRA EL PARKINSON identificada con Carnet de Extranjería número: 000756817 autorizo a Virginia del Pilar Monroy Malca con documento de identidad número: 74043491 a proporcionar mi nombre y número de teléfono, en mi calidad de Presidenta de la ASOCIACION ACTIVATE AREQUIPA LIGA CONTRA EL PARKINSON a fin de que personas ajenas a la Asociación busquen asesoramiento e información sobre la Enfermedad de Parkinson.

Atentamente.



Christine Allison Jeyachandran

CE: 000756817

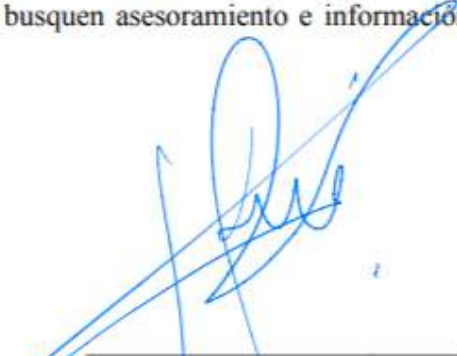
Anexo 10: Autorización para proporcionar nombre y teléfono – Koni Mejía

Arequipa, 02 de diciembre de 2020

Autorización:

Yo Koni Katerin Mejía Rojas, neuróloga del Hospital Nacional Daniel Alcides Carrion, identificada con DNI: 43280792 autorizo a Virginia del Pilar Monroy Malca con documento de identidad número: 74043491 a proporcionar mi nombre y número, en mi calidad de neuróloga, a fin de que las personas busquen asesoramiento e información sobre la Enfermedad de Parkinson.

Atentamente.



Koni Katerin Mejía Rojas

Anexo 11: Validez Despliegue

FICHA DE VALIDACIÓN DE DESPLIEGUE

Título del Proyecto

Diseño de un sistema de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona

Objetivo General: Implementar un modelo de visión computacional para el pre-diagnóstico de la enfermedad de Parkinson a partir de la escritura de una persona.

Nombre del investigador: Virginia del Pilar Monroy Malca

Nombre del experto: Koni Katerin Mejía Rojas

Especialidad: Neurología

Instrucciones: Determinar si el despliegue del modelo de predicción reúne los indicadores señalados y evaluar en la escala de Excelente, muy bueno, bueno, regular y malo.

N°	Indicadores	Definición	Excelente	Muy bueno	Bueno	Regular	Malo
1	Claridad y precisión	El despliegue es sencillo para los usuarios .	X				
2	Validez	El despliegue cumple con las preguntas necesarias sobre los principales síntomas del Parkinson de acuerdo con la OMS y el proceso de recolección de datos cumple el criterio de confidencialidad y reserva de datos del usuario.		X			
3	Organización	Las arquitecturas son adecuadas, comprenden desde el saludo, información, prueba en línea, obtención de datos del usuario.	X				
4	Confiabilidad	El despliegue es confiable porque el modelo de predicción tiene una buena métrica de precisión y es un buen proceso para obtener datos y aplicar una prueba de Parkinson.		X			
5	Extensión	El número de preguntas para tomar el examen de escritura no es excesivo y está en relación con las variables, dimensiones e indicadores del problema. Asimismo, todo el proceso de interacción entre el asistente virtual y el usuario no es excesivo.		X			
6	Inocuidad	El despliegue no constituye riesgo para el usuario.	X				

En consecuencia, el despliegue puede ser utilizado y está listo para empezar con la aplicación de test y recopilación de datos.



Firma del experto

DNI: 43280792

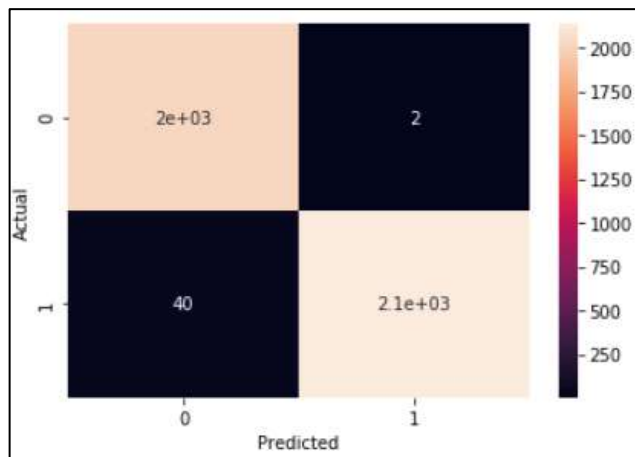
Para ver todo el detalle de resultados, se pueden descargar de:
<https://github.com/virginiamonroy/Tesis-VMM-2020>

ANEXO 12: Matriz de Confusión para la Metodología 3

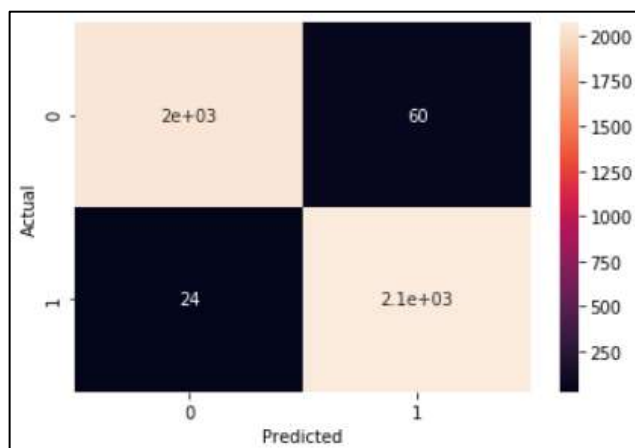
Base de datos de Brasil:

MEANDROS:

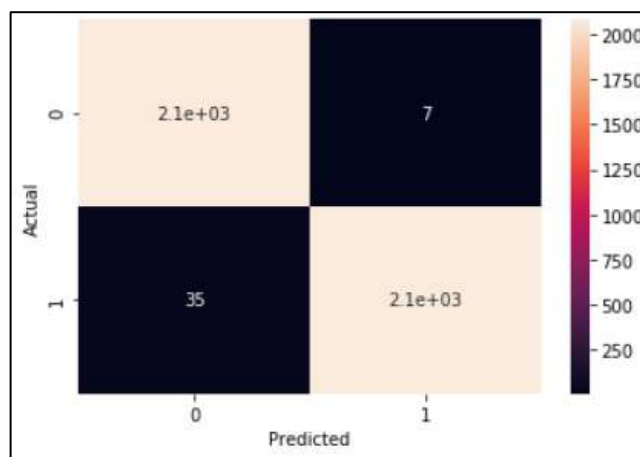
- VGG16
 - SVM



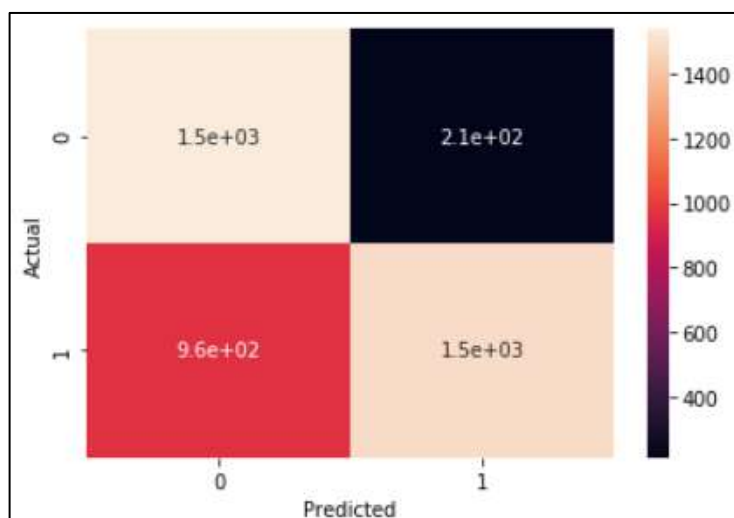
- RF



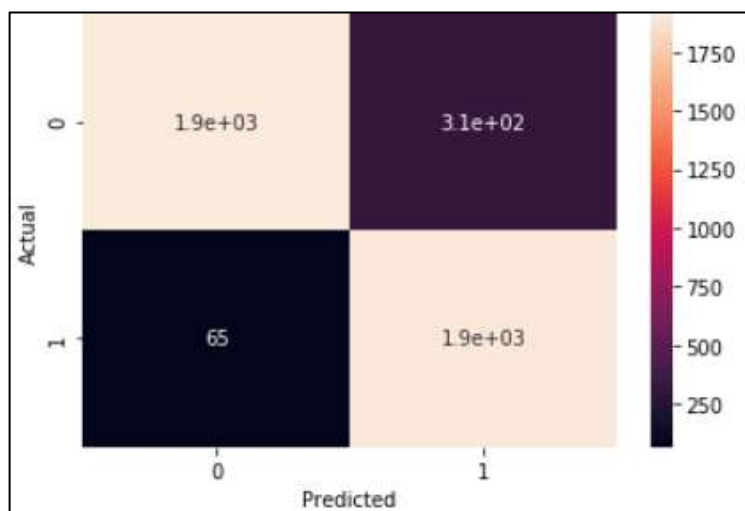
- KNN



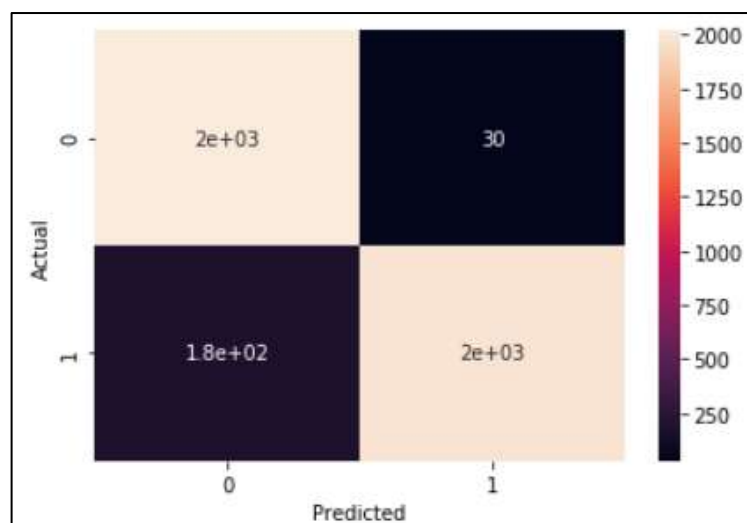
- VGG19
 - SVM



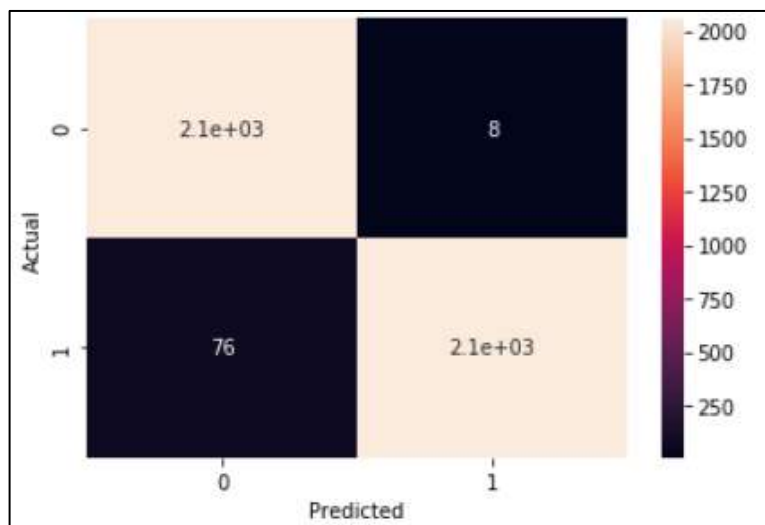
- RF



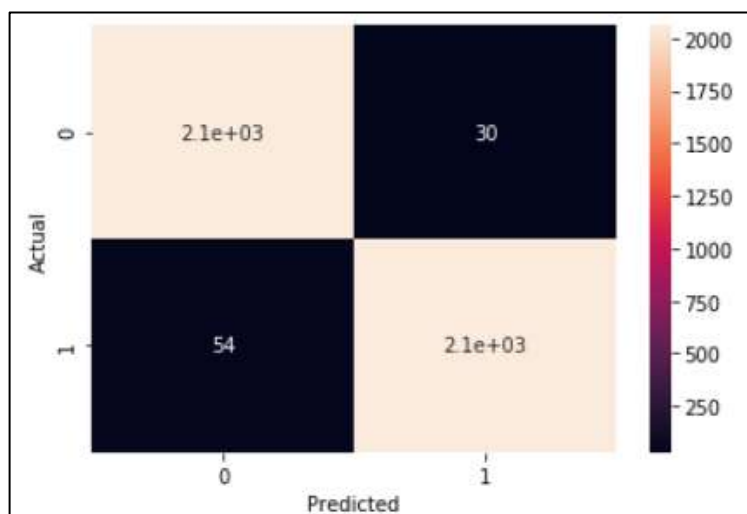
- KNN



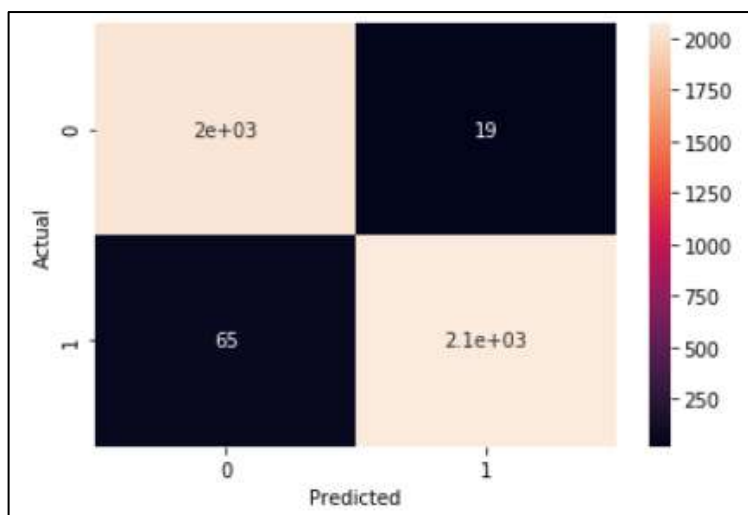
- Resnet
 - SVM



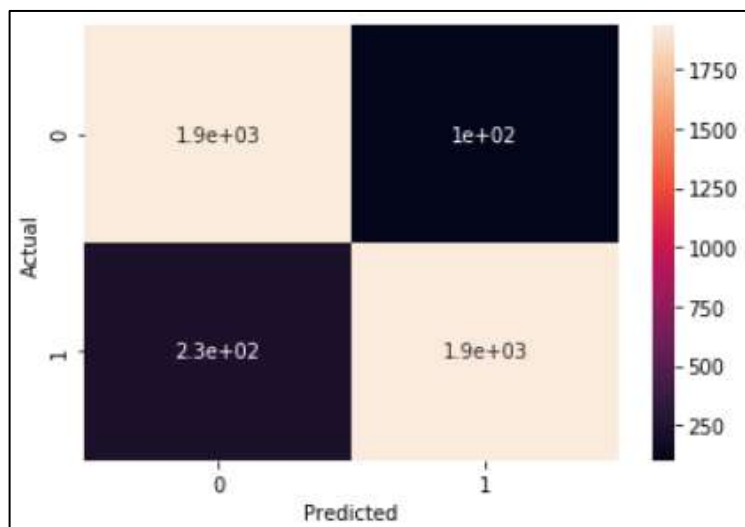
- RF



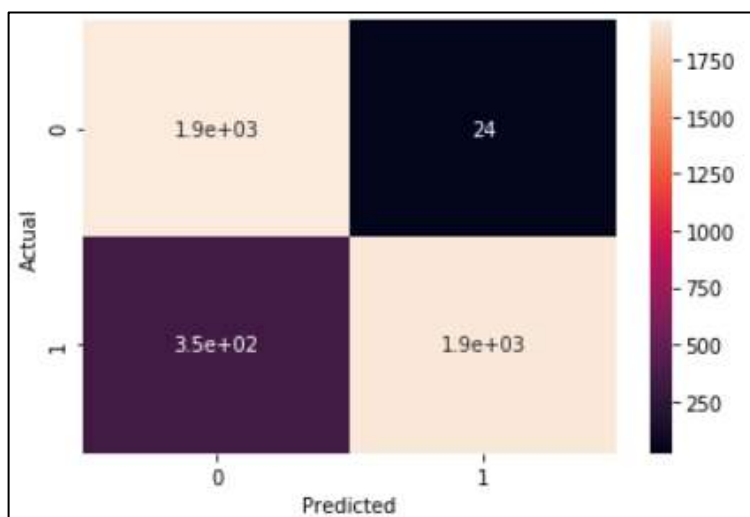
- KNN



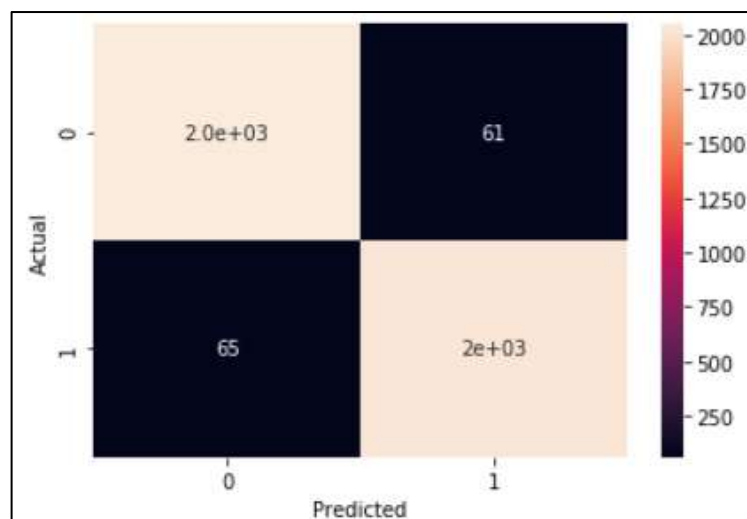
- Inception
 - SVM



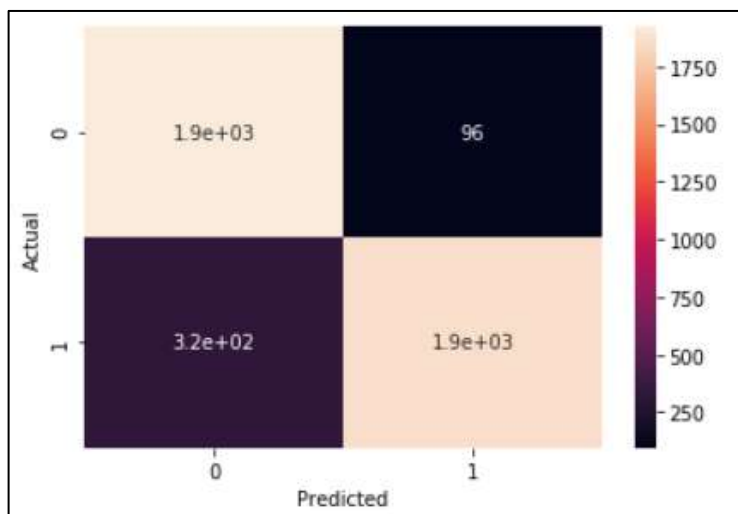
- RF



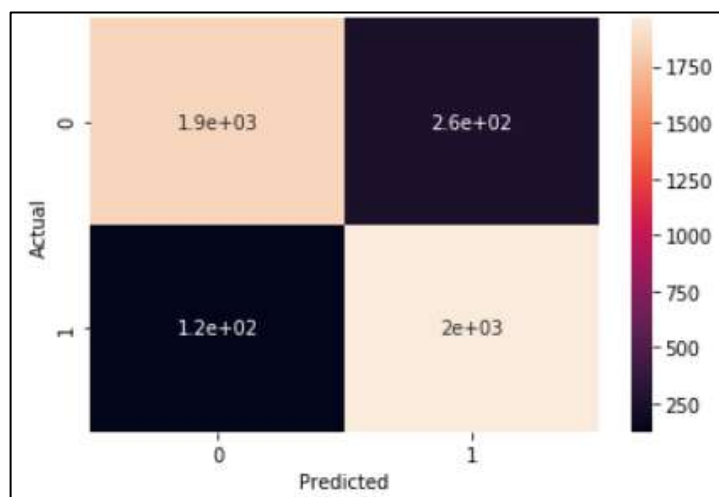
- KNN



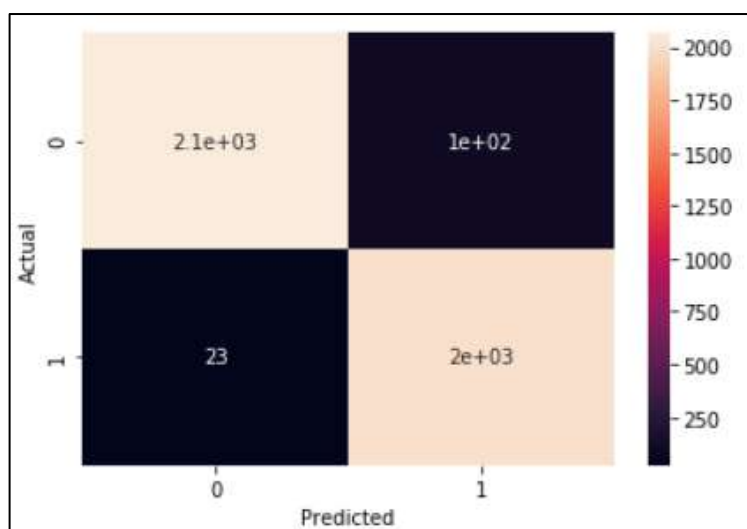
- Lenet
 - SVM



- RF

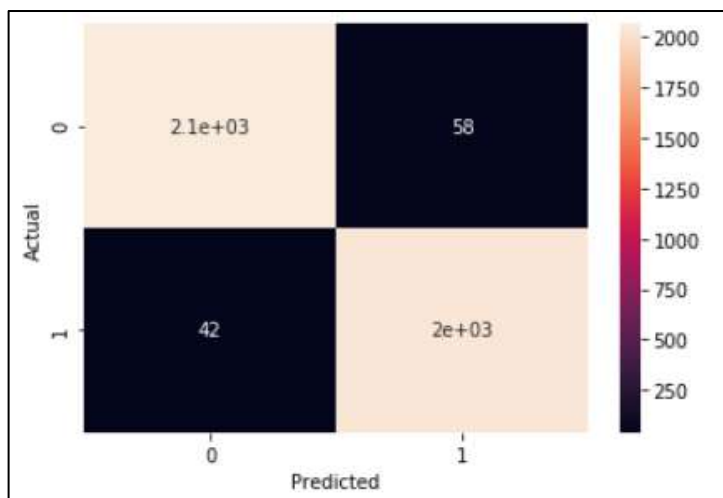


- KNN

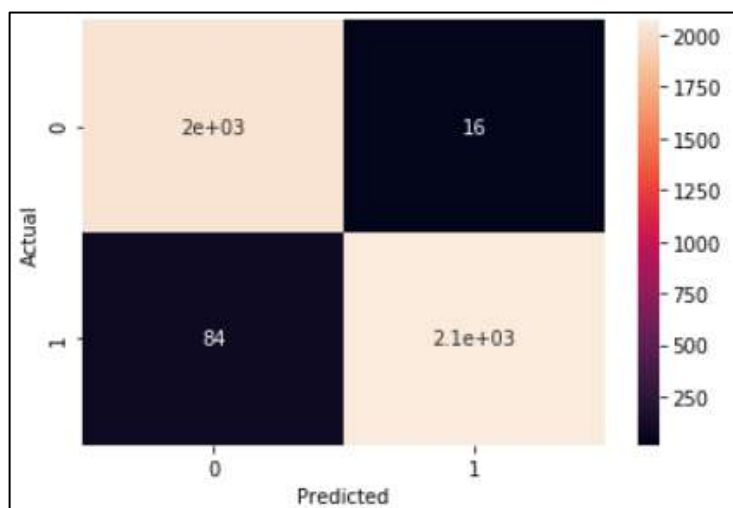


ESPIRALES:

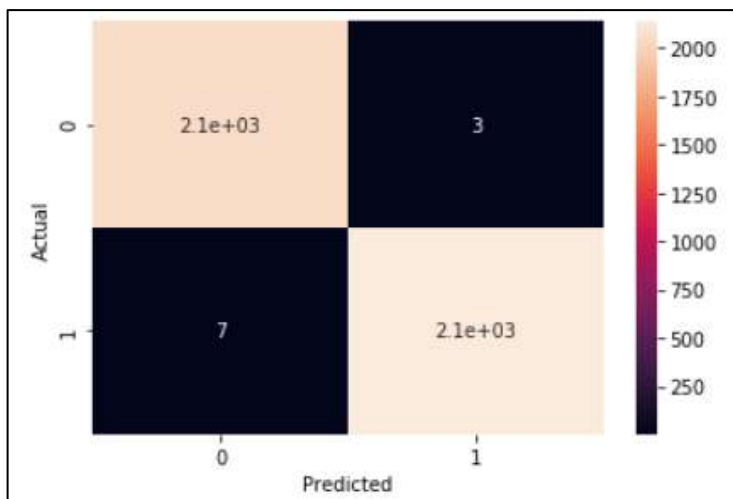
- VGG16
 - SVM



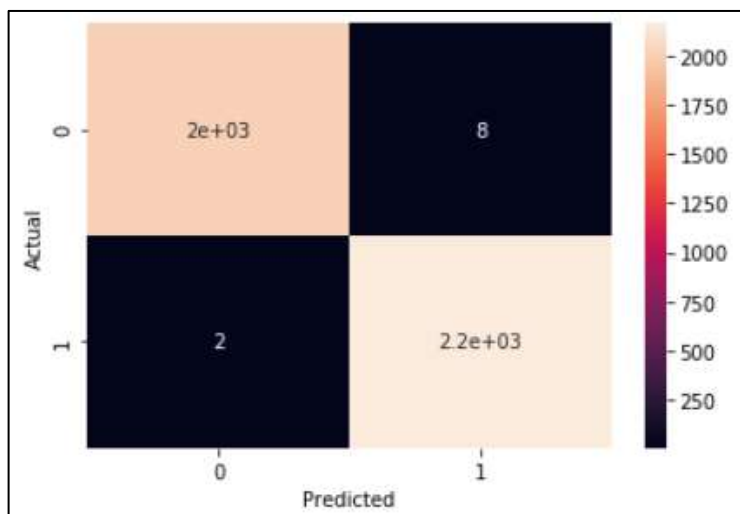
- RF



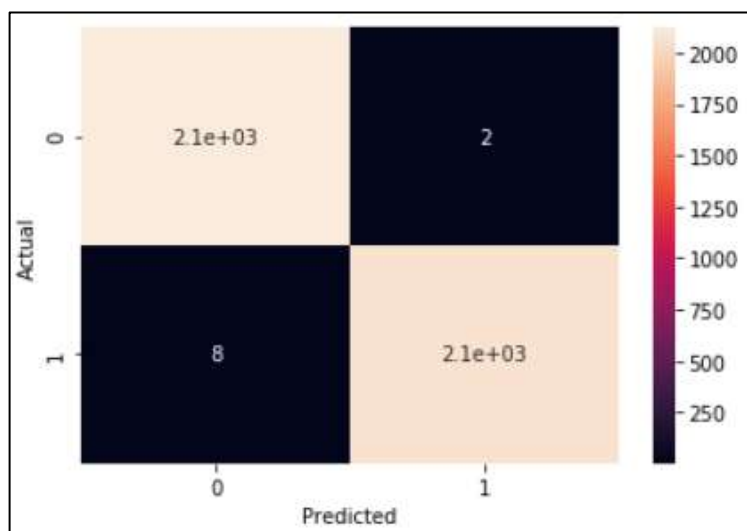
- KNN



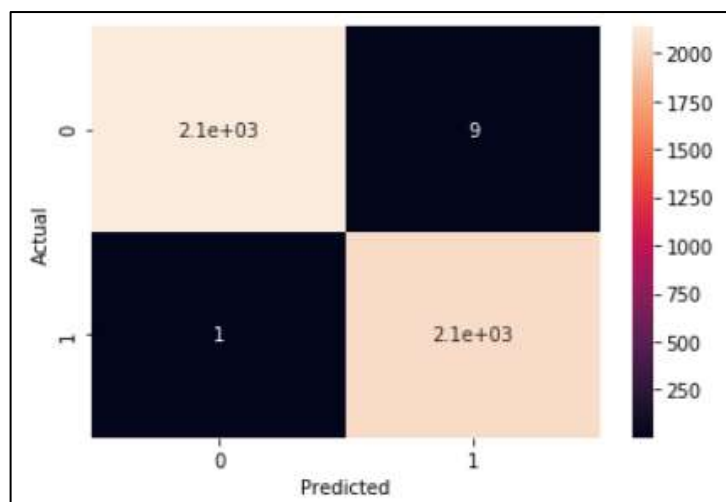
- VGG19
 - SVM



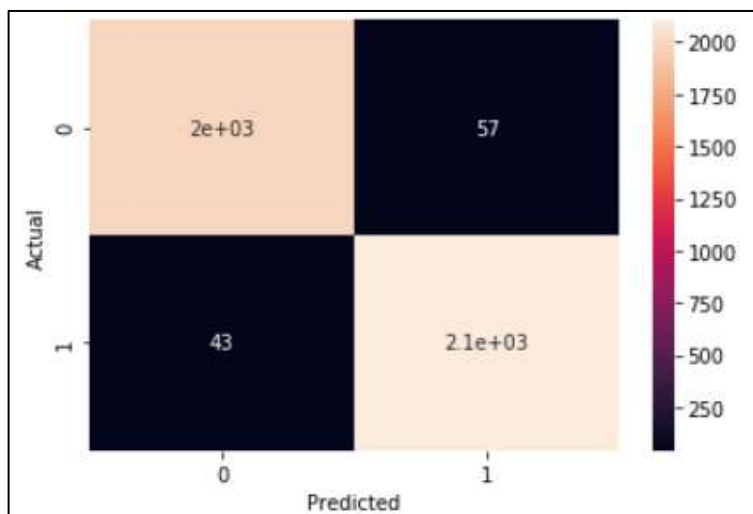
- RF



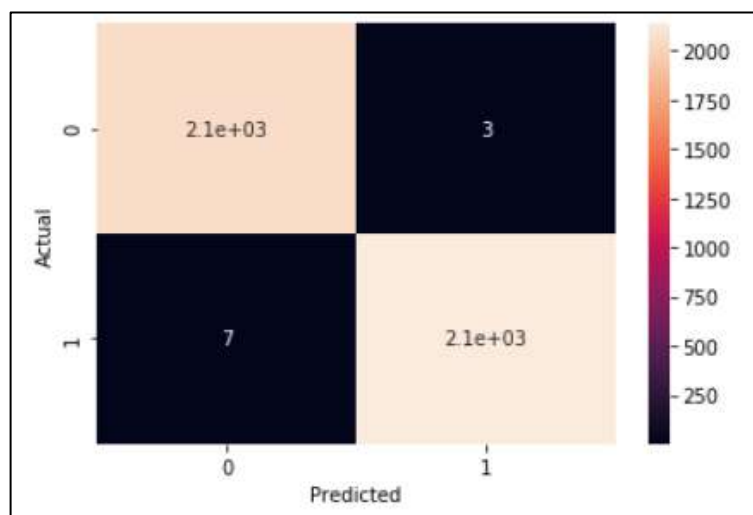
- KNN



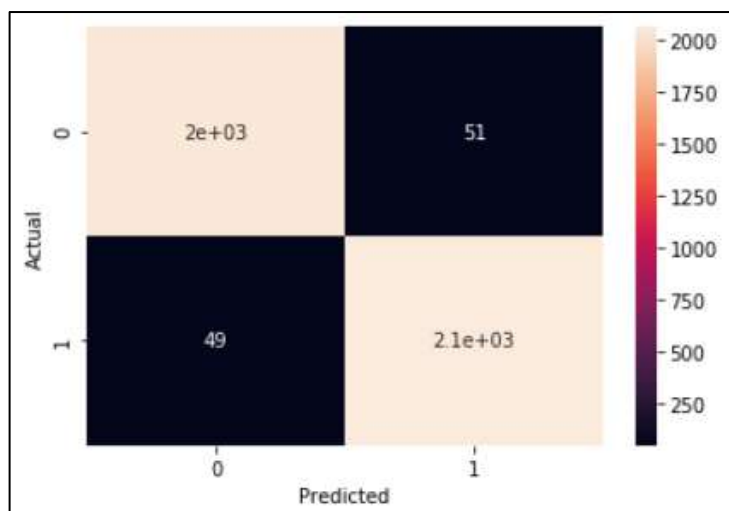
- Resnet
 - SVM



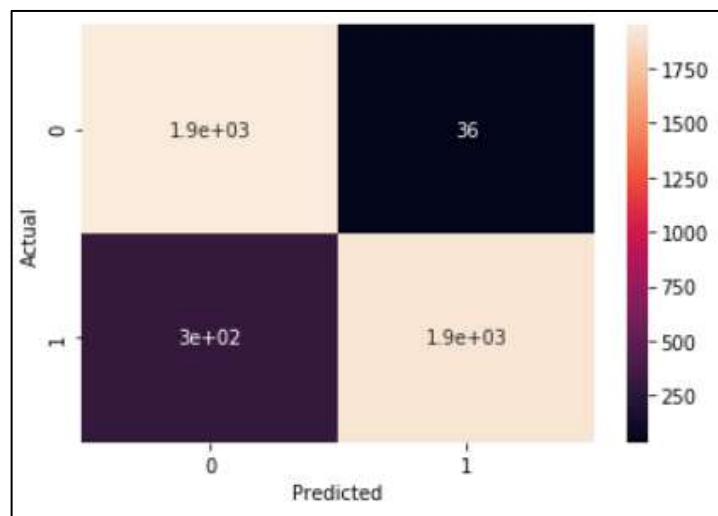
- RF



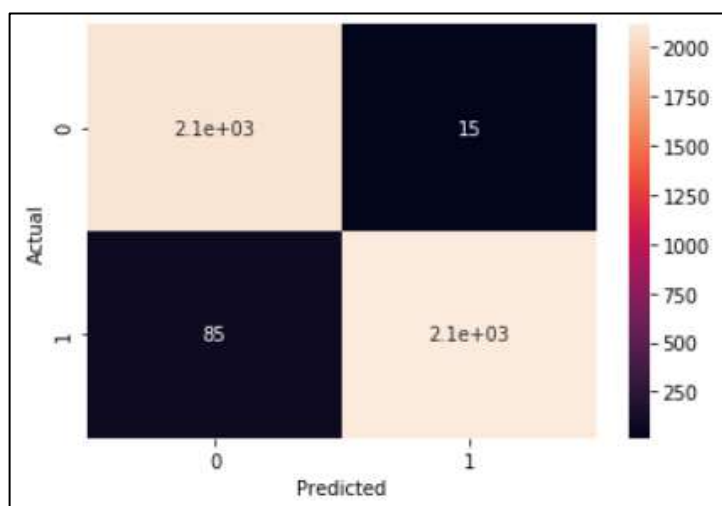
- KNN



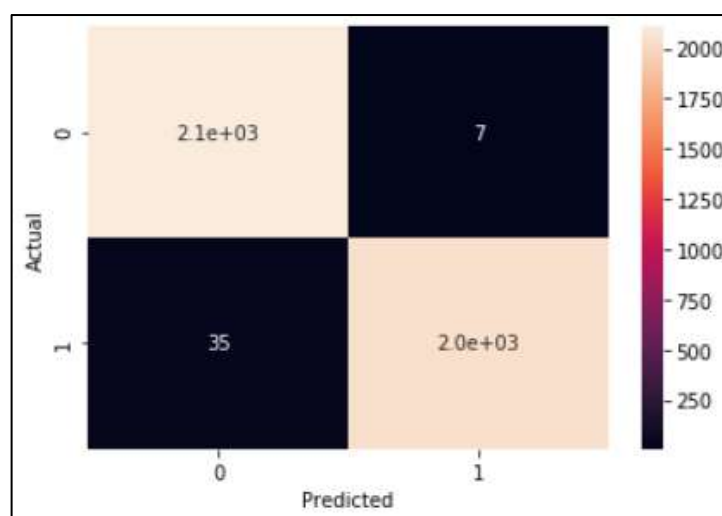
- Inception
 - SVM



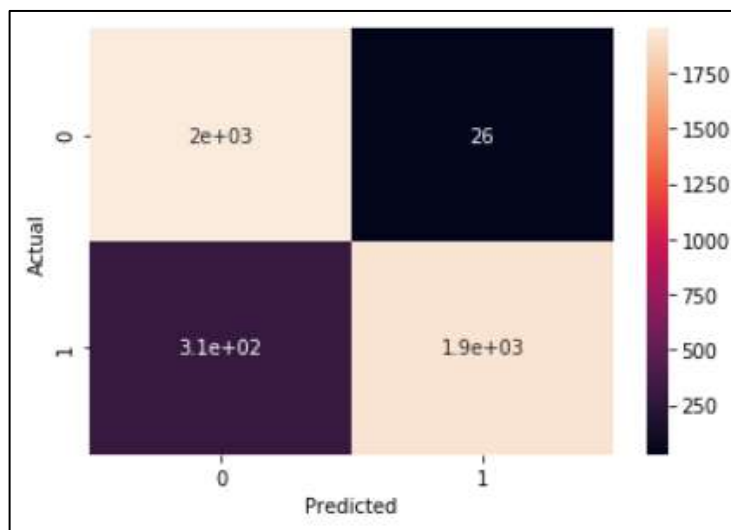
- RF



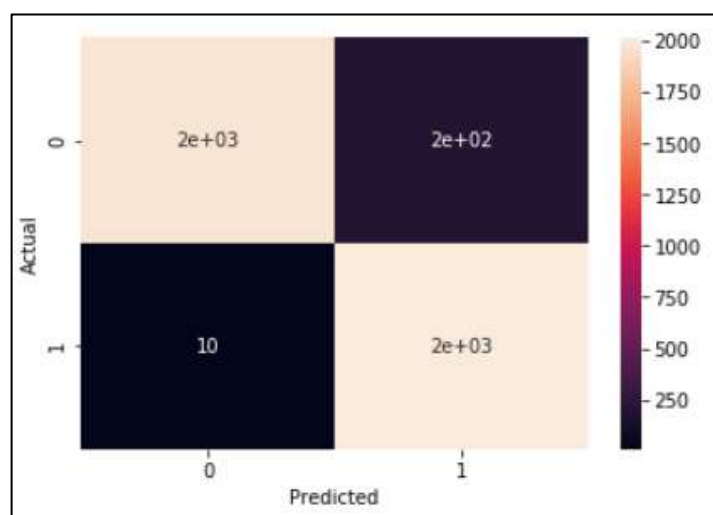
- KNN



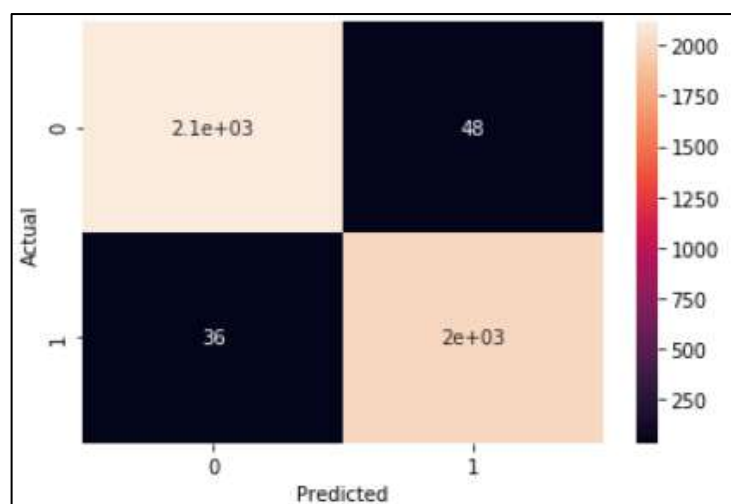
- Lenet
 - SVM



- RF



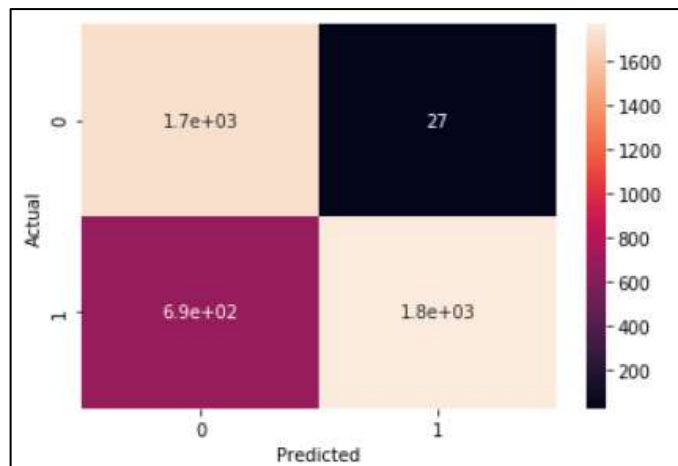
- KNN



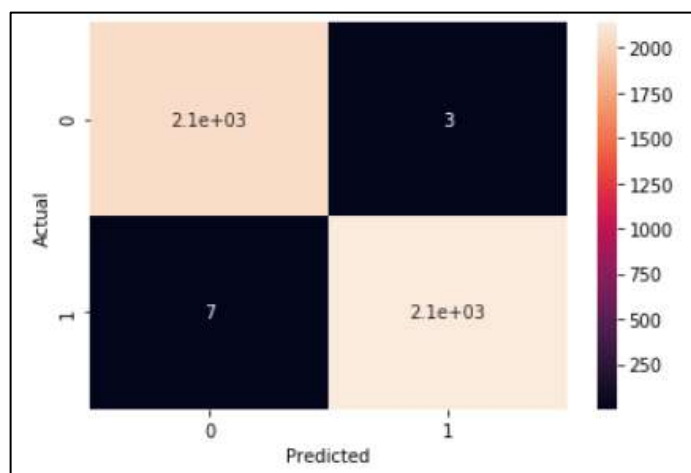
Base de datos de Perú:

MEANDROS:

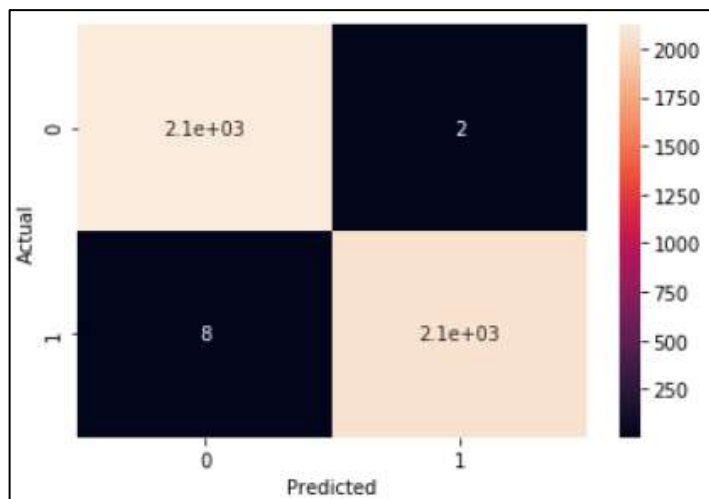
- VGG16
 - SVM



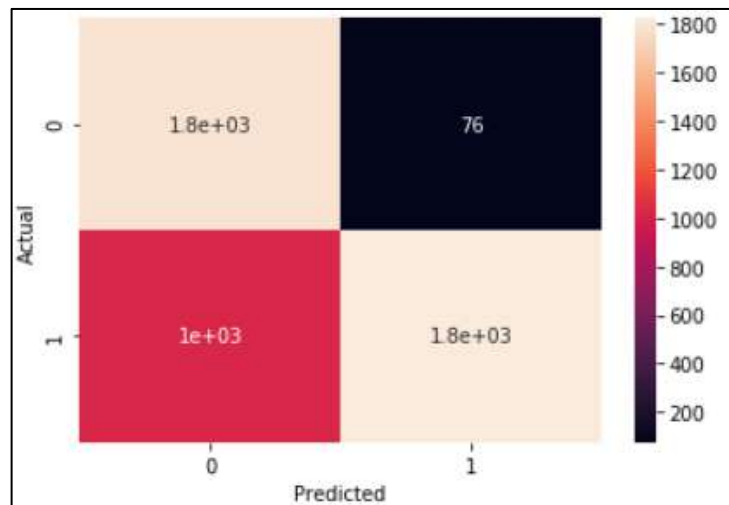
- RF



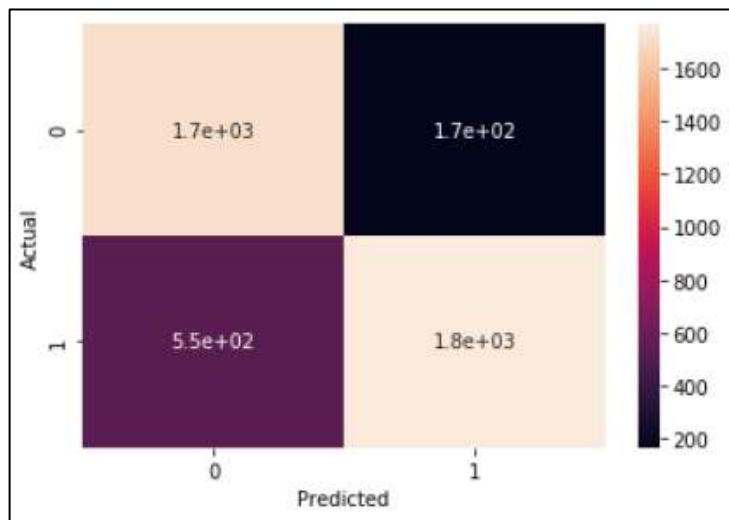
- KNN



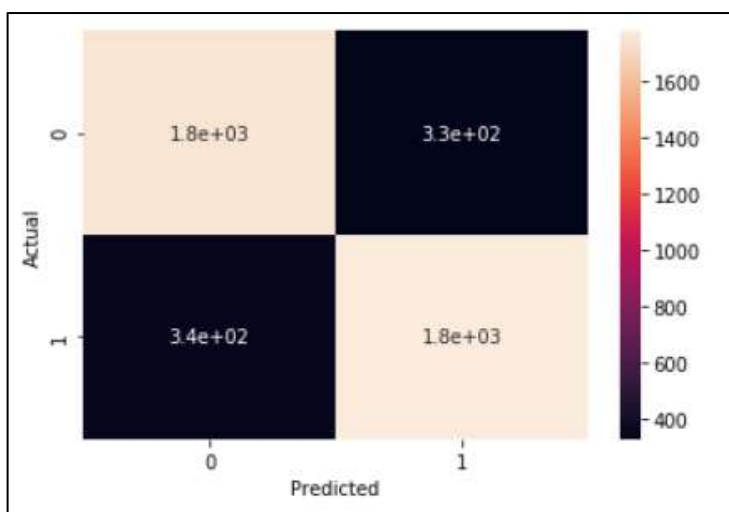
- VGG19
 - SVM



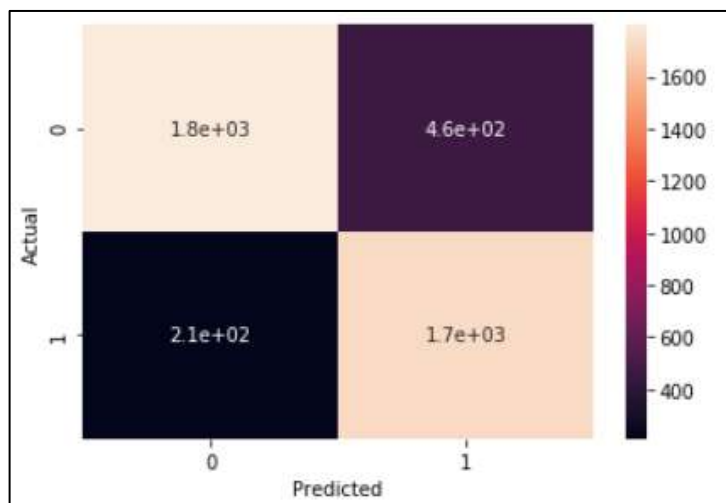
- RF



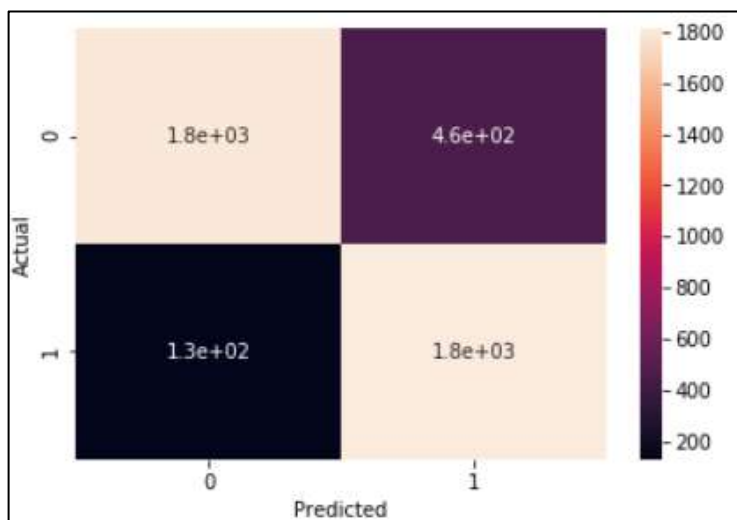
- KNN



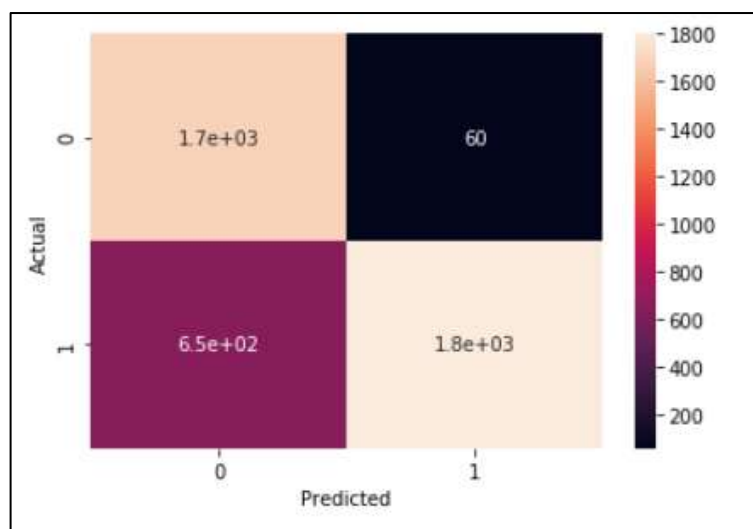
- Resnet
 - SVM



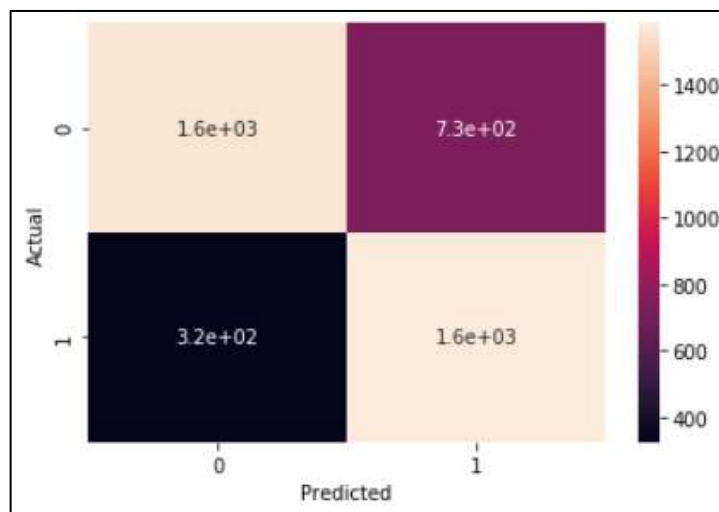
- RF



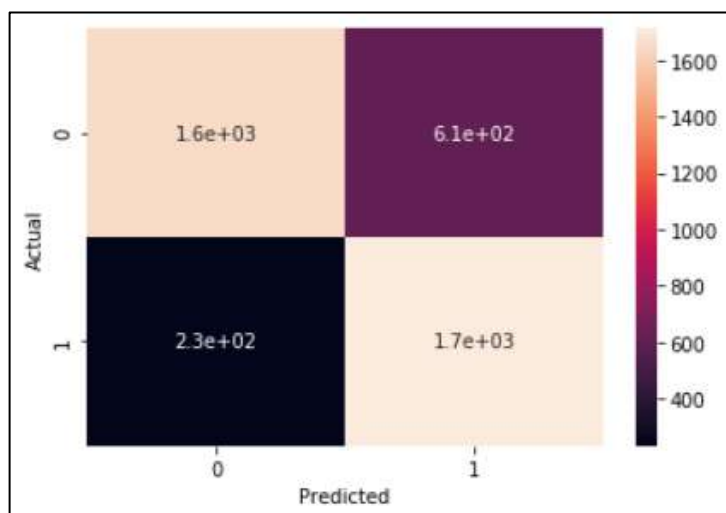
- KNN



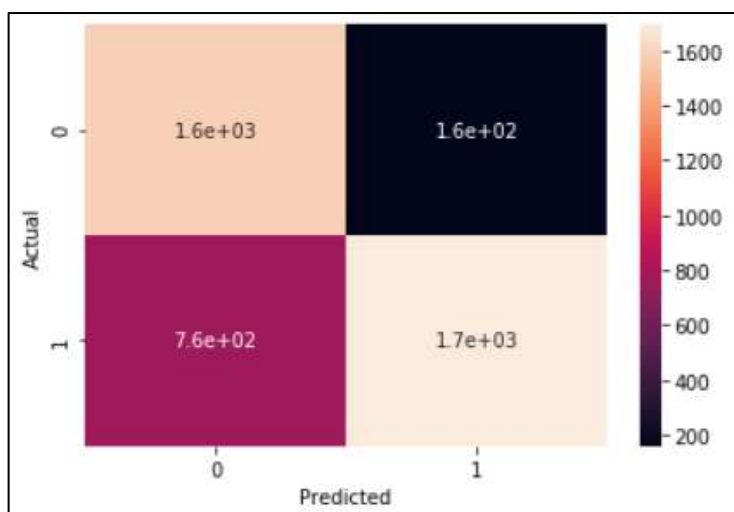
- Inception
 - SVM



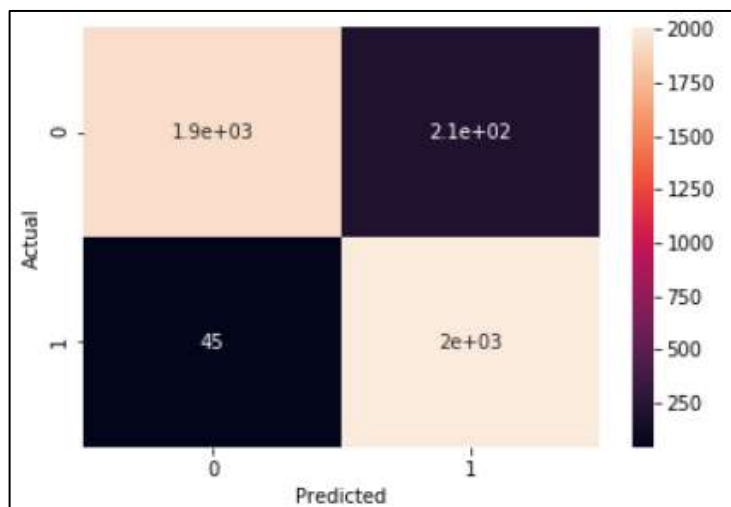
- RF



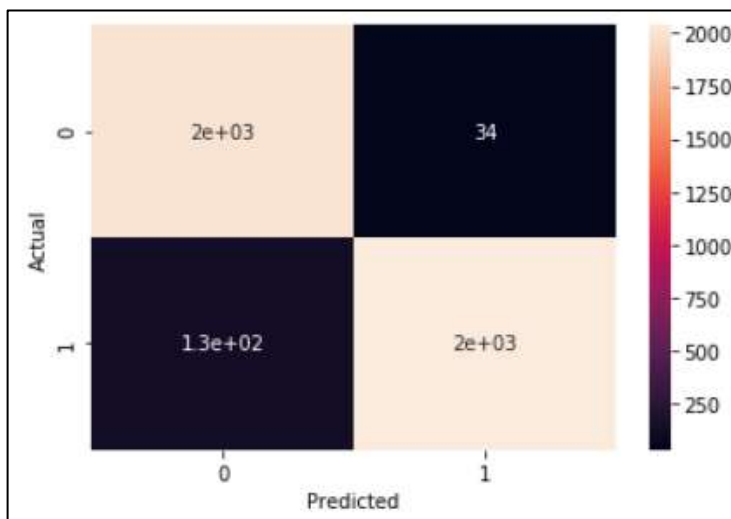
- KNN



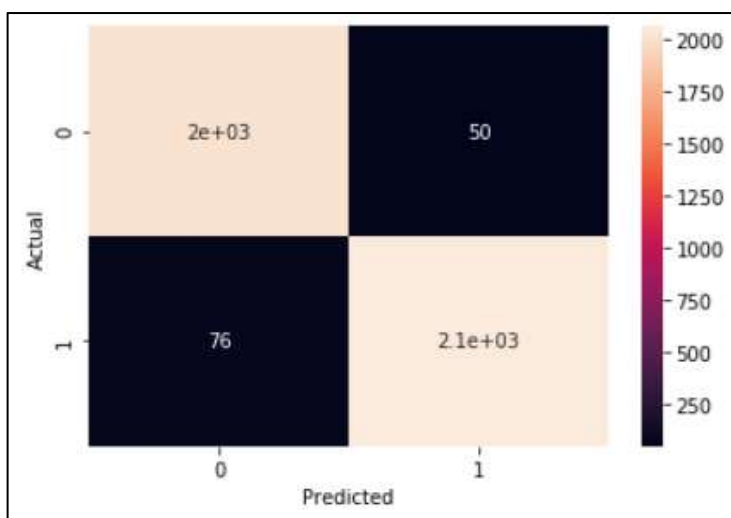
- Lenet
 - SVM



- RF

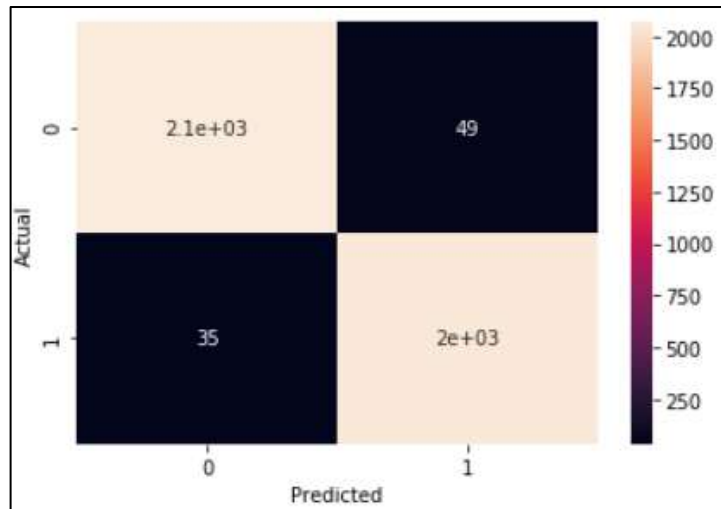


- KNN

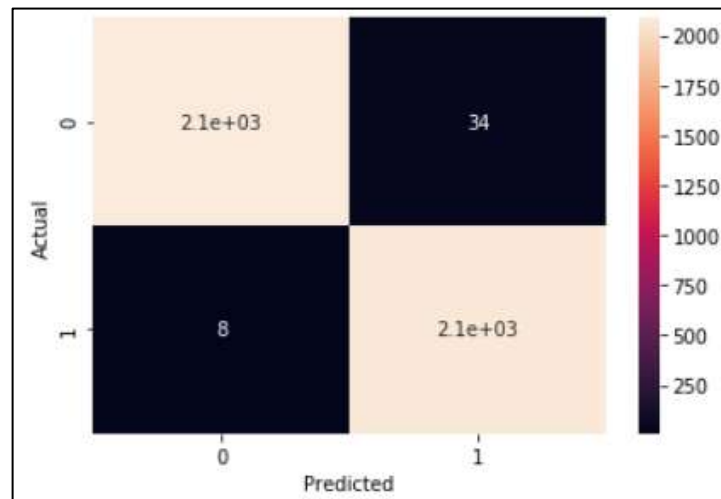


ESPIRALES:

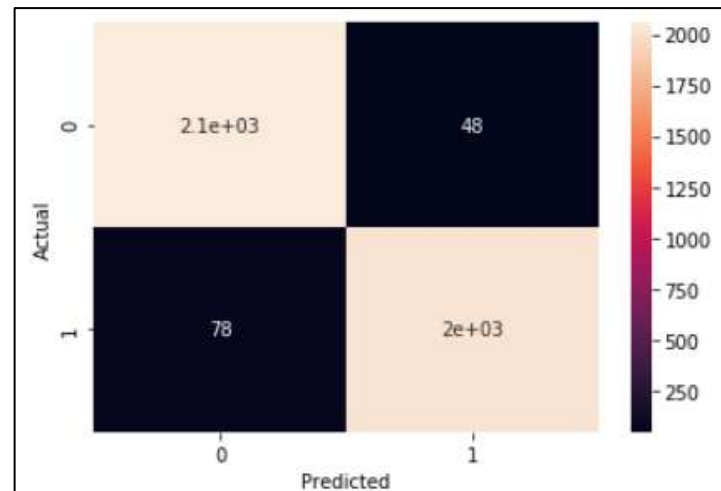
- VGG16
 - SVM



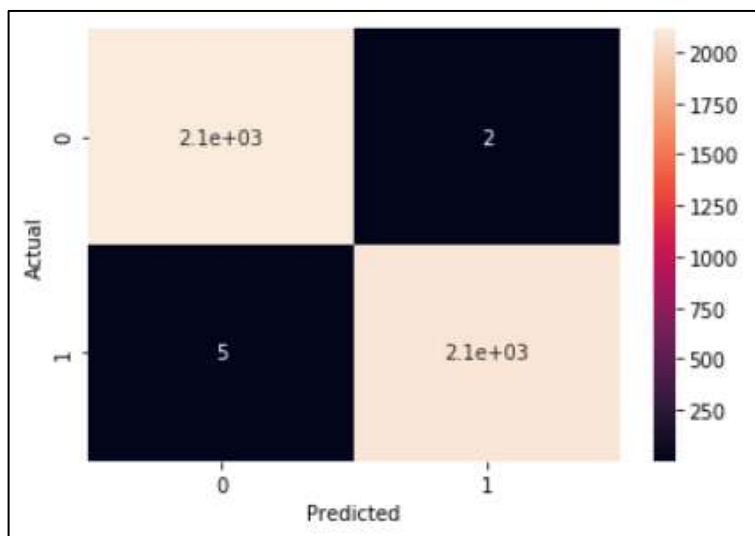
- RF



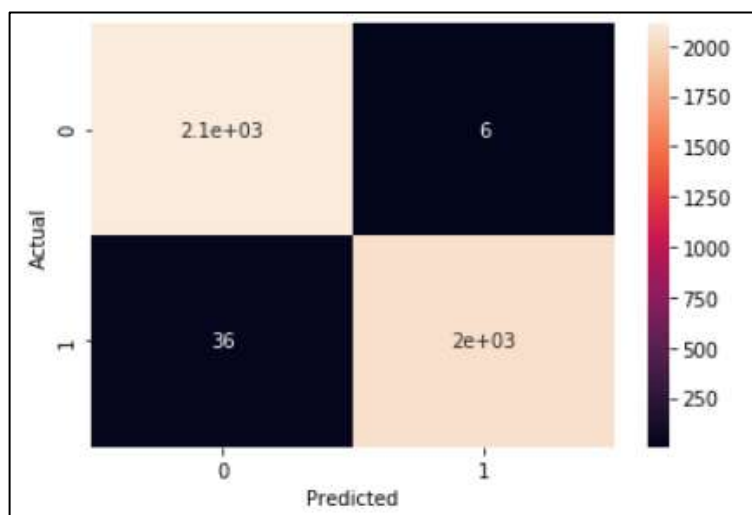
- KNN



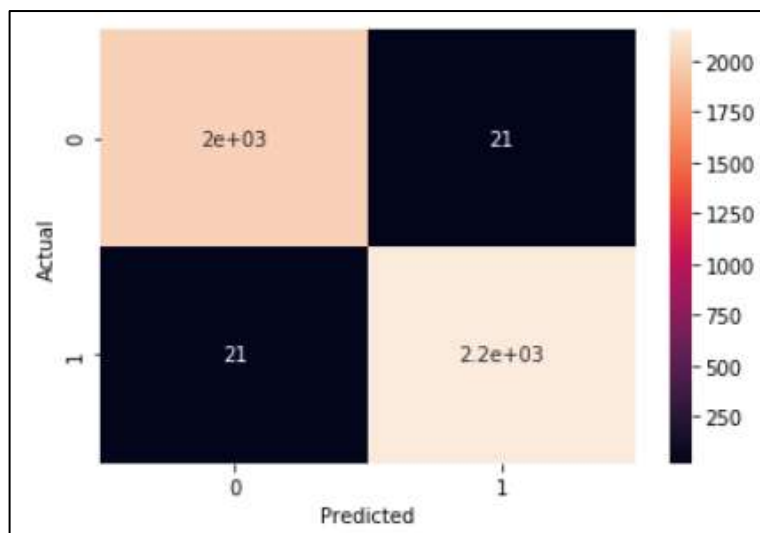
- VGG19
 - SVM



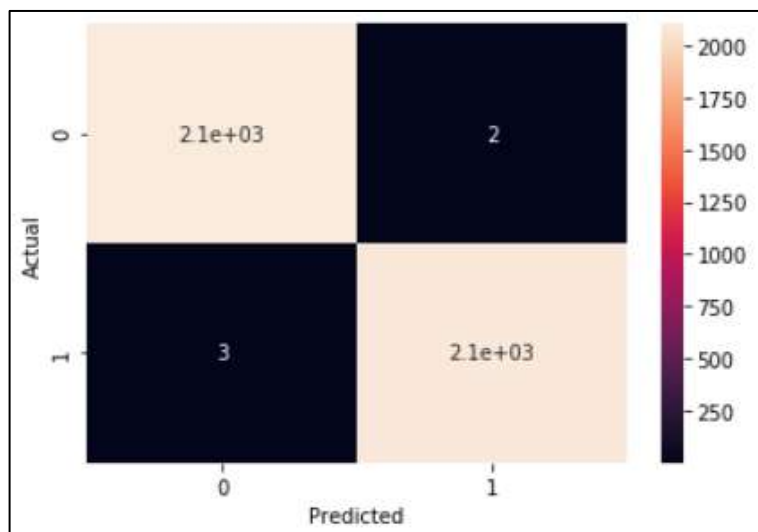
- RF



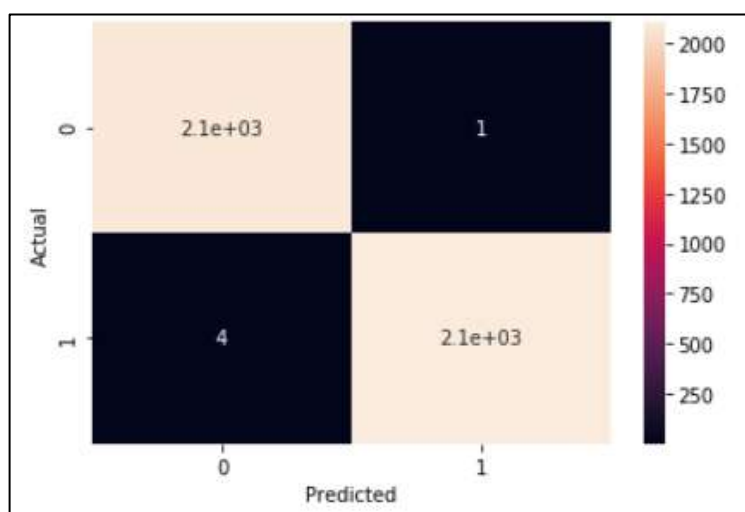
- KNN



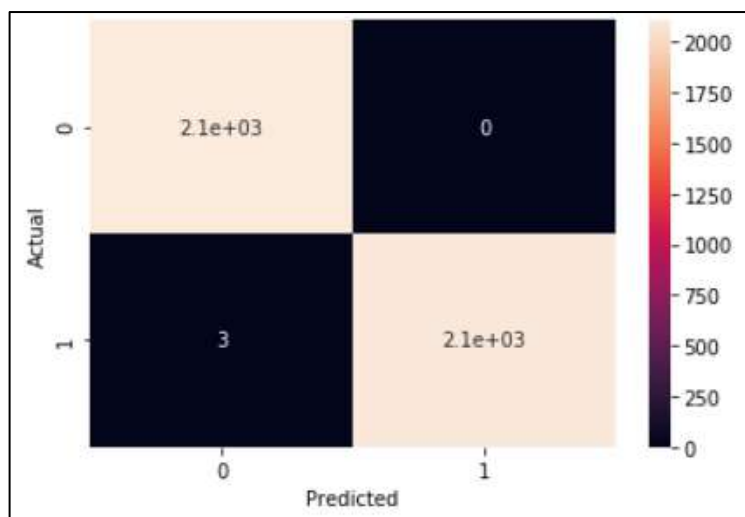
- Resnet
 - SVM



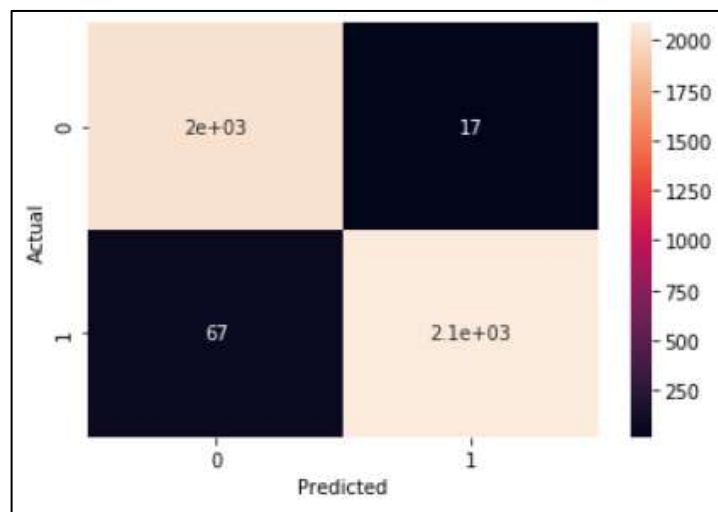
- RF



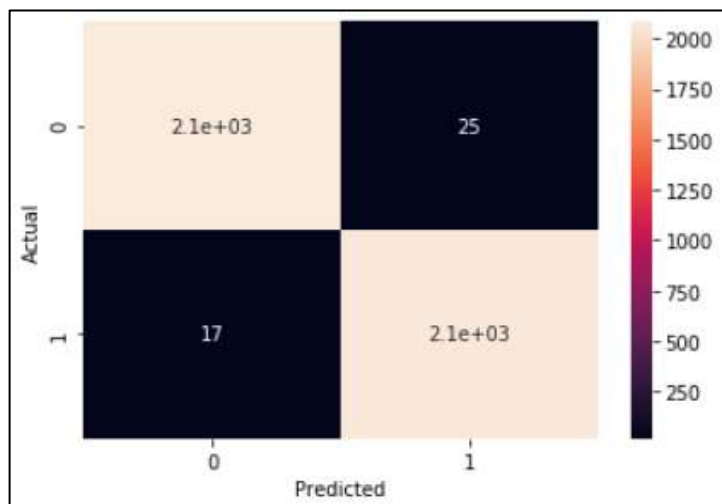
- KNN



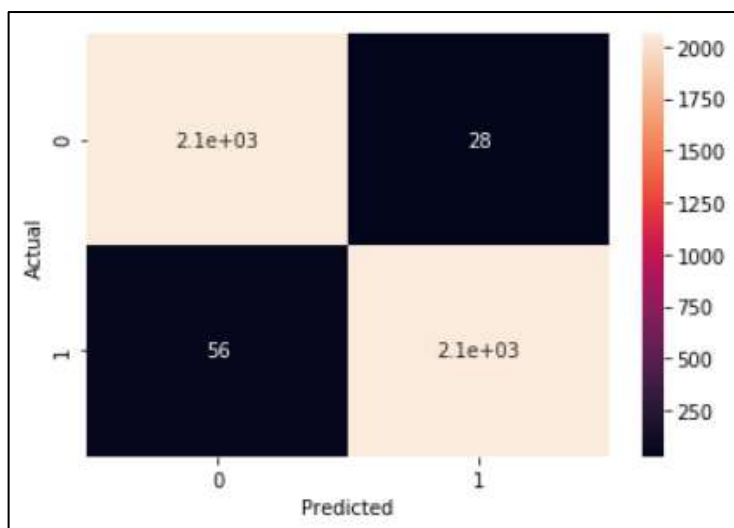
- Inception
 - SVM



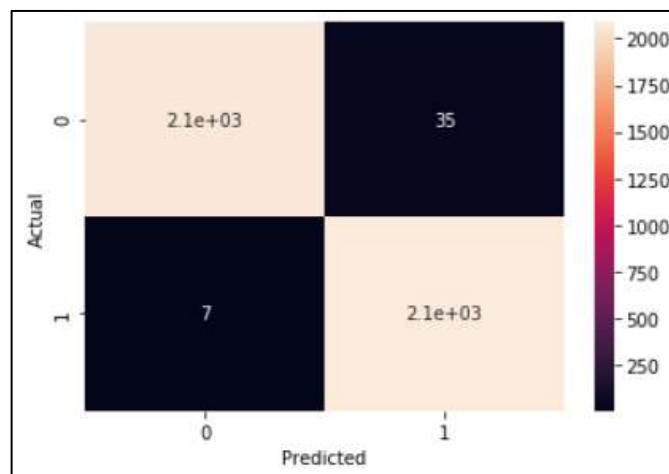
- RF



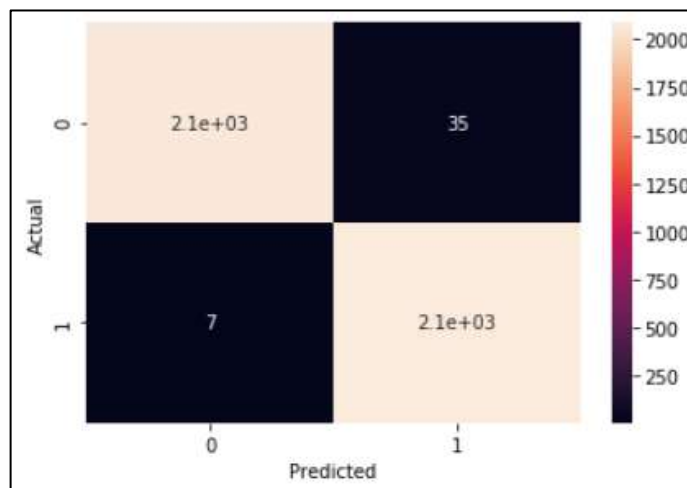
- KNN



- Lenet
 - SVM



- RF



- KNN

